

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra počítačové grafiky a interakce

## Procedurální modelování listů a simulace jejich stárnutí

**Filip Harant**

Školitel: Ing. Jaroslav Sloup  
Květen 2022



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Harant** Jméno: **Filip** Osobní číslo: **491863**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra počítačové grafiky a interakce**  
Studijní program: **Otevřená informatika**  
Specializace: **Počítačové hry a grafika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Procedurální modelování listů a simulace jejich stárnutí**

Název bakalářské práce anglicky:

**Procedural modeling of leaves and simulation of their aging**

Pokyny pro vypracování:

Seznamte se s problematikou procedurálního modelování listů a simulací morfologických a barevných změn vlivem stárnutí, vysychání a ročního období [1-4]. Vytvořte interaktivní aplikaci, která umožní generovat listy různých tvarů/typů a následně simulovat jejich morfologické změny. Pro simulaci morfologických změn využijte síť žilek listu vygenerovanou metodou [1]. Uživatelské rozhraní aplikace rozšiřte o možnost změny všech parametrů ovlivňujících chování implementovaných metod.

Funkčnost implementace ověřte na alespoň třech tvarově odlišných typech listů a dosažené výsledky porovnejte s reálnými fotografiemi.

Implementaci proveďte v C/C++ s využitím OpenGL.

Seznam doporučené literatury:

[1] Daeyeoul Kim, Jinmo Kim: Procedural Modeling and Visualization of Multiple Leaves. Multimedia Systems, vol.23, no.4, p.435-449, Springer, 2017.

[2] Xiaomin Wang, Chunjiang Zhao, Shenglian Lu, Xinyu Guo: Survey on Modeling and Visualization of Plant Leaf Color. Third International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications, pp. 417-424, 2009.

[3] SoHyeon Jeong, Si-Hyung Park, Chang-Hun Kim: Simulation of Morphology Changes in Drying Leaves. Computer Graphics Forum, vol.32, no.1, p. 204-215, Blackwell Publishing Ltd, 2013.

[4] Xiaopeng Sun, Jia Fu, Teng Chen, Yu Dong. Wrinkle and Curl Distortion of Leaves Using Plant Dynamic. Graphical Models, vol.118, Article 101118, Elsevier, November 2021.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Jaroslav Sloup Katedra počítačové grafiky a interakce**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **01.02.2022**

Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. Jaroslav Sloup  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.  
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## Poděkování

Chtěl bych poděkovat své rodině za poskytnuté zázemí, neutuchající podporu během studií a důvěru ve mě. Rád bych také poděkoval svému vedoucímu práce Ing. Jaroslavu Sloupovi za jeho velkorysost a ochotu si na mě udělat čas kdykoliv jsem potřeboval s něčím poradit či cokoliv probrat.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že na mou bakalářskou práci se vztahují práva a povinnosti stanovené zákonem č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 20. května 2022 ...

## Abstrakt

Tato bakalářská práce se věnuje morfologii listů, procesu generování listů pomocí žilek a jejich procedurálními deformacemi. Pro tyto účely jsou popsány dosavadní existující metody generování listů a deformací, na jejichž základě je vypracováno řešení, které popsané metody spolu kombinuje. Vygenerované listy a vyprodukované deformace jsou na závěr vizualizovány v OpenGL.

**Klíčová slova:** morfologie listů, generování listů, deformování listů, procedurální deformace, vizualizace, žíly, stárnutí, vysychání, osmóza, osmotický tok vody, OpenGL, CImg, C++

**Školitel:** Ing. Jaroslav Sloup  
Praha 2, Karlovo náměstí 13, E-413

## Abstract

This bachelor thesis deals with the morphology of leaves, the process of generating leaves using veins and their procedural deformations. For these purposes, the existing methods of generating leaves and deformations are described, on the basis of which a solution is developed that combines the described methods. The generated sheets and produced deformations are finally visualized in OpenGL.

**Keywords:** leaf morphology, leaves generating, leaves deforming, procedural deformations, visualization, veins, aging, drying, osmosis, osmotic water flow, OpenGL, CImg, C++

**Title translation:** Procedural modeling of leaves and simulation of their aging

# Obsah

<b>1 Úvod</b>	<b>1</b>
1.1 Cíl bakalářské práce a její struktura .....	2
<b>2 Související práce</b>	<b>5</b>
2.1 Modelování listů a simulace procesu růstu pomocí implicitní funkce .....	6
2.2 Procedurální modelování a vizualizace listů pomocí metody generování žilek .....	8
2.3 Simulace morfologických změn u vysychajících listů za použití manuálně zadaných žilek .....	12
2.4 Deformace listů pomocí dynamiky rostlin založené na hlavních žilách.	17
<b>3 Návrh řešení a realizace</b>	<b>23</b>
3.1 Generování segmentů a konfigurační soubor .....	23
3.2 Růst hlavní žíly .....	24
3.3 Laterální a sublaterální žíly ....	25
3.4 Detekce překročení žilek za oblast kontury .....	27
3.5 Generování profilu podle konfiguračního souboru .....	28
3.6 Struktura listu a inicializace hmoty .....	29
3.7 Osmóza vody a deformace .....	32
<b>4 Konkrétní implementace</b>	<b>37</b>
<b>5 Výsledky</b>	<b>39</b>
<b>6 Diskuse</b>	<b>49</b>
<b>7 Závěr</b>	<b>51</b>
<b>Literatura</b>	<b>53</b>
<b>A Obsah přiloženého média</b>	<b>55</b>

## Obrázky

1.1 Ukázka skladby vrstvy listu . . . . .	2	3.2 Předešlý obrázek znázorňující průběh růstu žíly . . . . .	25
1.2 Příklady reálných listů, které se budeme snažit vymodelovat a deformovat . . . . .	3	3.3 Vygenerované hlavní žíly . . . . .	25
1.3 Příklady reálných vyschlých listů, kterým se budeme chtít deformací vymodelovaného listu přiblížit . . . . .	3	3.4 Vygenerované hlavní, laterální a sublaterální žíly . . . . .	26
2.1 Fáze vývoje listu . . . . .	6	3.5 Vygenerované hlavní, laterální a sublaterální žíly . . . . .	27
2.2 Ukázka parametru $t_{\text{Switch}}$ řídicího fáze vývoje listu . . . . .	6	3.6 Listy podle kontury . . . . .	27
2.3 Ukázka vektorového pole . . . . .	7	3.7 Spojení počátečního a vygenerovaného cílového segmentu nalézajících se uvnitř kontury listu, které vystupuje z kontury . . . . .	28
2.4 Ukázka vývoje listu . . . . .	8	3.8 Profil listu . . . . .	29
2.5 Detekce a aproximace kontury listu za použití této metody . . . . .	9	3.9 Rozmístění částic žil (zelená) a částic povrchu (černá). . . . .	30
2.6 Získané kandidátní body a růst žíly	9	3.10 Dvouvrstvá struktura vizualizovaná na obrázku dubu. . . . .	31
2.7 Procesy detekce kandidátních bodů na segment a růst žil v závislosti na kontuře . . . . .	10	3.11 Částice žil spolu tvoří trojúhelník, jinak by počáteční distribuce hmoty nebyla správná a neprobíhal by správný osmotický tok vody. . . . .	32
2.8 Výsledky růstu hlavních a laterálních žil dosažených touto studií . . . . .	11	3.12 Dvě různé vizualizace rozložení hmoty . . . . .	32
2.9 Ukázka vygenerovaných bodů a vzniklých vlasovitých žilek . . . . .	12	3.13 Hmotnost jednotlivých částic delaynovy vrstvy . . . . .	34
2.10 Proces modelování . . . . .	13	3.14 Pohyb jednotlivých částic delaynovy vrstvy a jejich hmotnost	35
2.11 Ukázka částic a pružin vrstev seshora a z boku . . . . .	13	5.1 List javoru a vygenerované žilky	39
2.12 Neuniformní rozložení hmoty s hodnotami mezi $(m_{\text{min}}, m_{\text{max}})$ . . . . .	14	5.2 List ořešáku a vygenerované žilky	39
2.13 Ukázka struktury hydratovaných a dehydratovaných útvarů v závislosti na difúzním koeficientu $\lambda$ . . . . .	15	5.3 List kaštanovníku jedlého a vygenerované žilky . . . . .	40
2.14 Simulace deformace listů s různě navolenými parametry . . . . .	17	5.4 List dubu a vygenerované žilky . . . . .	40
2.15 Vývoj algoritmu . . . . .	18	5.5 Porovnání vygenerovaných listů líčidla amerického s reálným listem	41
2.16 Ukázka buněk a jejich struktury	18	5.6 Porovnání vygenerovaných listů javoru s reálným listem . . . . .	41
2.17 Systém hmota-pružina . . . . .	19	5.7 Porovnání vygenerovaných listů dubu s reálným listem . . . . .	42
2.18 Obrázek popisující vypětí a pohyb částice $p_i$ . . . . .	20	5.8 Porovnání vygenerovaného listu kaštanovníku jedlého s reálným listem . . . . .	42
2.19 Ukázka procesu postupné deformace listů s využitím této metody (převzato z: [5]). . . . .	21	5.9 Deformace listů vygenerované metodou založené na hlavních žilkách . . . . .	42
3.1 Znázornění nalezené kontury obrázku listu s černým pozadím pomocí červené barvy . . . . .	24	5.10 Deformace listů vygenerované metodou založené na žilkách . . . . .	43



5.11 Ukázka námi vygenerovaných deformací líčidla amerického . . . . .	44
5.12 Ukázka námi vygenerovaných deformací listu javoru . . . . .	45
5.13 Ukázka námi vygenerovaných deformací listu dubu . . . . .	46
5.14 Ukázka námi vygenerovaných deformací listu kaštakovníku . . . . .	47

## Tabulky

5.1 Parametry vymodelovaných listů a počet částic a pružin . . . . .	43
--	----



# Kapitola 1

## Úvod

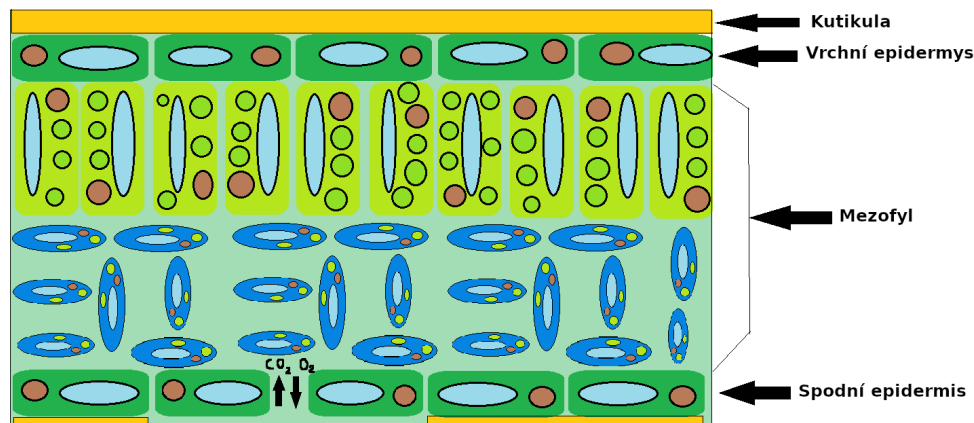
S postupným vývojem technologií, jako je virtuální realita (VR), počítačová grafika a další, rostou i nároky na rozumnou, přesnou a fyzikálně věrohodnou vizualizaci objektů. Zaměříme-li se na vizualizaci a modelování scén, jedním z nejfrekventovaněji vyskytujících se objektů je listí. V některých případech je list využit jako jeden z hlavních objektů, častější je však jeho použití jako pozadí, kdy se podílí na tvorbě atmosféry scenerie, vytváří určitý tématický nádech a může tím ovlivnit i náladu pozorovatele. To vše dělá z listí velice žádanou komoditu. Vezmeme-li však v potaz, že ideálně bychom chtěli, aby každý list vypadal jinak a měl mírně odlišné vlastnosti, vyžadovala by se tím od animátora extrémní trpělivost při modelování či animacích a vymodelovat celou scénu kousek po kousku by mohlo být velice zdouhavé a těžkopádné. Z tohoto důvodu bylo v minulosti navrženo několik metod, které se snažily tento proces generování listů, a následně jejich deformací, zautomatizovat.

Existují metody pro generování listů, jež přišly s rozumnými výsledky pro generování listů za použití implicitní funkce nebo vytváření žil, a metody pro deformace, jež přišly s přijatelnými výsledky při deformacích. Tyto metody však vygenerované listy z ostatních metod pro generování listů nebraly v úvahu a svůj model použitý pro deformace si vytvořily svým vlastním jednoduchým způsobem.

Abychom docílili reálných výsledků, je nutné při deformacích dodržet zákony fyziky. Co se listů týče, generování žilek, které bylo použito v této práci, nemusí bez ztráty funkčnosti zohledňovat jakékoliv fyzikální zákony. Pro účely vizualizace a generování žilek nám postačí referenční obrázek, na kterém si můžeme ověřit, jak moc jsme se skutečnosti přiblížili. Naopak při deformacích, kdy používáme námi vygenerované žilky, zákony fyziky už uplatníme, jelikož chceme simulovat morfologické změny listu, což nezbytně vyžaduje vědět, jak proces stránutí, vysychání a deformace funguje.

Povrch listu je tvořen buňkami, které nazýváme epidermis (viz obr.1.1). Jedná se o jednovrstvé krycí pletivo, které zajišťuje nejen ochranu rostliny před vnějšími vlivy, ale především rostlinu chrání před ztrátou vody, reguluje výměnu  $O_2$  a  $CO_2$  a vylučuje některé metabolity. Nejsvrchnější vrstvu listu tvoří kutikula. Jedná se o nebuněčnou vrstvu, která chrání rostlinu před slunečním zářením a nadměrným výparem vodních par či je bariérou pro různé patogeny. Základním pletivem listu je mezofyl. Jedná se o buňky

obsahující chloroplasty podílející se na fotosyntéze. Zároveň slouží jako hlavní buňky pro výměnu plynů a vody. Nejspodnější vrstvu tvoří spodní epidermis, který navíc obsahuje průduchy, které zajišťují výměnu plynů mezi listem a okolím.



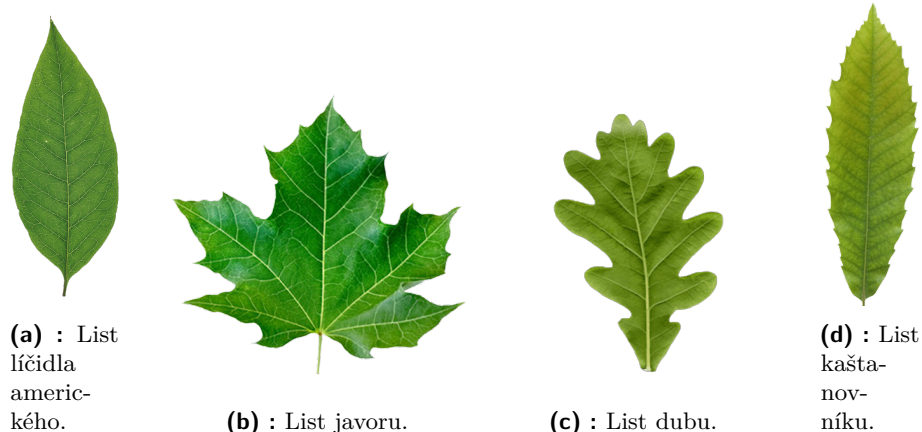
**Obrázek 1.1:** Ukázka skladby vrstvy listu podle bakalářské práce [10].

Mezofylové buňky jsou v této práci generovány nejprve společně s žilkami, později jsou generovány před modelováním deformací listů při tvorbě delaunayovy triangulace, aby byl získán povrch listu, na kterém bychom mohli simulovat morfologické změny. Jelikož 70 % hmotnosti buňky tvoří voda, můžeme simulováním jejího úbytku dosáhnout deformací buňky. Míra deformací je ovlivněna množstvím vody, kterým daná buňka disponuje. Jak buňka vysychá vlivem slunečního záření, poměr koncentrace živin a vody se v ní mění, buňky si mezi sebou začínají vyměňovat vodu, dokud se koncentrace živin a vody nevyrovná. Jev, kdy se tekutina šíří mezi buňkami přes polopropustnou membránu, nazýváme osmóza. Polopropustná membrána, která osmózu umožňuje, nepropouští všechny živiny, nýbrž je propustná pouze pro tekutiny. Vhodným simulováním osmózy jsme schopni provést reálnou výměnu vody mezi buňkami.

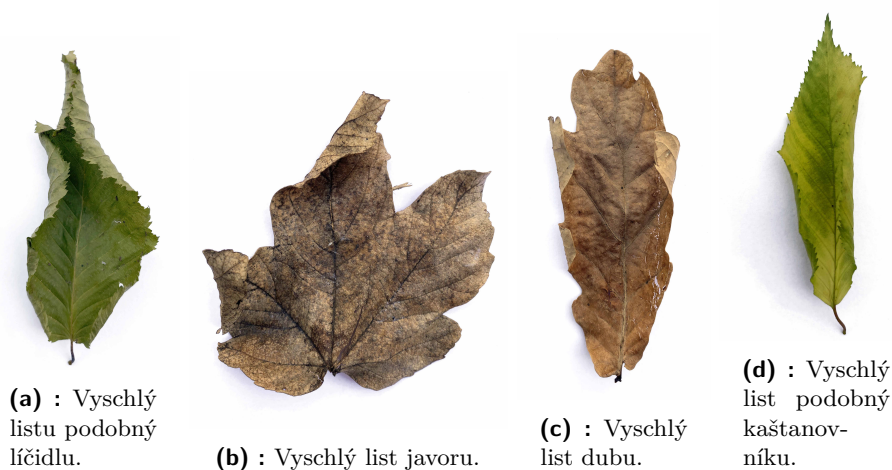
Jak list vlivem stárnutí postupně vysychá, deformují se nejprve buňky (částice) nacházející se poblíž hrany listu. To je zapříčiněno tím, že nejvíce živin a vody obsahují částice žíly, která funguje také jako jejich hlavní transportér. Čím větší je vzdálenost částice povrchu listu od žíly, tím menší má hmotnost a tím více je náchylná na vysychání a deformace.

## 1.1 Cíl bakalářské práce a její struktura

V této práci navrhujeme metodu na intuitivní generování žilek mnoha různých tvarů listů a jejich deformaci, kde cílem práce je za použití existujících metod vygenerovat žilky a simulovat jejich růst. Takto vygenerované žilky zkombinovat s procesem deformací a docílit co nejrealističtějších výsledků podobným reálným listům, jako jsou listy na obr.1.2 a deformacím na obr. 1.3.



**Obrázek 1.2:** Příklady reálných listů, které se budu snažit vymodelovat a deformovat.



**Obrázek 1.3:** Příklady reálných vyschlých listů, kterým se budeme chtít deformací vymodelovaného listu přiblížit.

Nejdříve se zaměříme na popis studií a ty nejvíce zvažované či použité popíšeme dopodrobna. Dále bude následovat návrh řešení, kde probereme, jak řešení vypadá a co z metod bylo použito či přidáno. Probereme generování dat listů pro část metody, kde generujeme žíly, její samotnou implementaci, a nakonec popíšeme proces deformace. Část generování žil se budeme snažit rozšířit o různé funkcionality, které se v uvedených studiích nevyskytovaly, jako je například rozšíření o prakticky neomezenou variabilitu listů použitím datového a konfiguračního souboru json. Tento soubor pak může zastávat roli knihovny pro různé typy listů a pomocí řady modifikátorů a volitelných parametrů ovlivňovat samotné rozmístění, počty, úhly, délkové profily, barvy a tloušťky žil, míru distribuce vody a živin, rozsahy náhodných faktorů a mnoho dalšího a tím téměř neomezeně ovlivňovat výsledné tvary libovolných listů, obdobně jako v biologickém světě rozmanitost druhů určuje DNA.



## Kapitola 2

### Související práce

Existuje několik prací, které se zabývají generováním listů od samotného počátku, nebo deformují již hotové listy. Žádná práce však dosud nespojovala proces generování listů a procedurálních deformací v takovém rozsahu. Budeme generovat a deformovat listy za pomoci žilek. Můžeme ji rozdělit na dvě části - část generování žilek a část deformací.

Dostupné studie popisující generování listů fungují každá odlišně a používají při tom každá jiné přístupy. Jedna ze studií [1], kterou budu popisovat, se zabývá generováním žilek, s jejichž pomocí probíhá rekonstrukce listu a vychází z ní i semestrální práce kolegy Martina Pažouta [2], který se touto prací inspiroval a jehož postup mi pomohl porozumět částem, které mi přišly obtížné. Generování žilek popisuje i studie od Alsweise [3], nicméně nedosahuje tak uspokojivých výsledků, jakých bychom si přáli dosáhnout. Nepoměrně přesvědčivější výsledky podává naopak studie [4], která se pokouší přímočaře generovat list pomocí natahování malého geometrického útvaru, který roste až do finální podoby určené uživatelem.

Naopak studie pojednávající o deformacích fungují na podobném principu a mají mnoho společného. Nejnovější práce pojednávající o deformacích [5] dosahuje velice sympatických výsledků, nicméně pro jejich dosažení používá pouze hlavní žíly. Laterální a sublaterální žíly nebere v úvahu a to by neumožňovalo metody správně napojit. Naproti tomu této práci předcházející studie od Jeonga [6] využívá všech žil pro počáteční inicializaci hmoty pomocí Bellman-Fordova algoritmu a umožňuje nám práci dokonale provázat.

Z prací generujících listy jsme si vypůjčili metodu pro generování žilek a její výstup následně propojili s metodou procedurálního generování deformací a oboje tak zkombinovali.

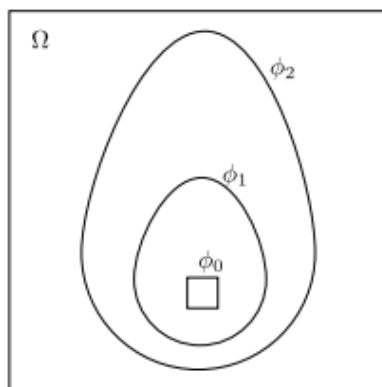
V této kapitole se zaměříme na popis jednotlivých souvisejících prací, popíšeme jejich princip a případně uvedeme důvod, proč byly či nebyly použity.

## 2.1 Modelování listů a simulace procesu růstu pomocí implicitní funkce

Modelování listů a simulace procesu růstu pomocí implicitní funkce [4] přichází s dvou-fázovou simulací pro dosažení cílového tvaru listu. Pro růst listu uvažuje tzv. znaménkovou funkci vzdáleností  $\Phi$ , která udává vzdálenost bodu do hrany listu. Tato funkce je proměnná v čase a lze ji vyjádřit následovně:

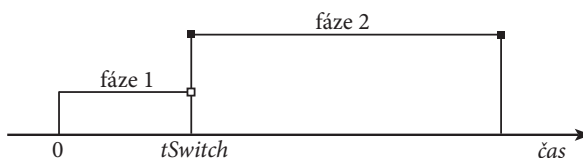
$$\Phi = \{(x, y) \in \Omega : \Phi(x, y) = 0\} \quad (2.1)$$

V této práci je generován tvar listu bez modelování jeho žilek. List prochází třemi fázemi  $\Phi_0$ ,  $\Phi_1$  a  $\Phi_2$ . Na začátku je list v tzv. primordiu  $\Phi_0$  (viz obr. 2.1), což je typicky malý kroužek či čtvereček, který je použit pro inicializaci. Tento objekt je postupně zvětšován v první fázi dokud nedosáhne velikosti definované referenčním obrázkem  $\Phi_1$ . Následuje fáze  $\Phi_2$ , během které je růst ovlivňován vektorovým polem, které list natahuje ve směrech jeho působení. Finální podoba listu nastává ve fázi  $\Phi_3$ , která je výsledkem ukončené fáze  $\Phi_2$ .



**Obrázek 2.1:** Fáze vývoje listu (převzato z: [4]).

Přechod mezi fázemi je řízen parametrem  $t_{\text{Switch}}$  (viz obr. 2.2), který udává, v jakém momentě má ke změně dojít.



**Obrázek 2.2:** Ukázka parametru  $t_{\text{Switch}}$  řídicího fáze vývoje listu (převzato z: [4]).

Pro první fázi vývoje je připraven transparentní obrázek listu, který je převeden do stupňů šedi a označen jako  $\mu_0$ . V momentě, kdy se budeme zabývat detekcí pomezí mezi listem a transparentním pozadím, bude nás zajímat gradient obrázku, který nám v důsledku velkého rozdílu mezi intenzitou barev



listu a pozadí vyjde velmi vysoký. Právě díky gradientu jsme posléze schopni detekovat hranu listu. Implicitní funkce  $\Phi$  tuto hranu reprezentuje v čase. Abychom docílili toho, že  $\Phi$  hranu v daný moment skutečně reprezentuje, musí být počáteční funkce  $\Phi_0$  definovaná uživatelem, která udává tvar primordia, velmi malá s jedinou podmínkou, že její lokace je uvnitř listu. Pro realističtější výsledky je doporučeno, aby pozice primordia byla v okolí řapíku listu.

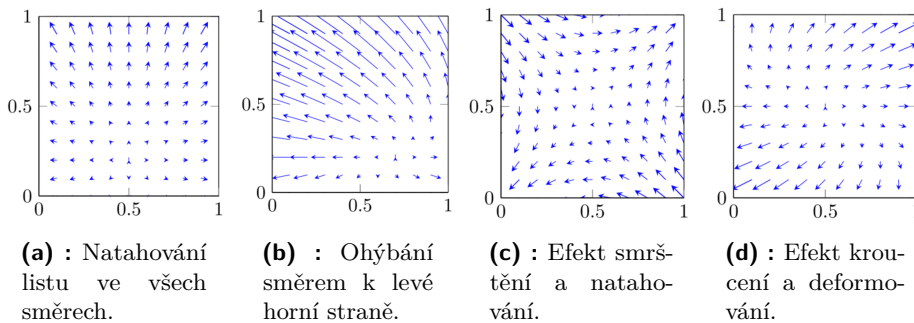
Ve druhé fázi je jako startovní bod použit výstup z první fáze, list je deformován a zvětšován pomocí vektorového pole (viz. obr. 2.3) a funkce vzdáleností  $\Phi$  je reinitializována podle vztahu:

$$\Phi_t + S(\Phi_0)(|\Delta\Phi| - 1) = 0 \quad (2.2)$$

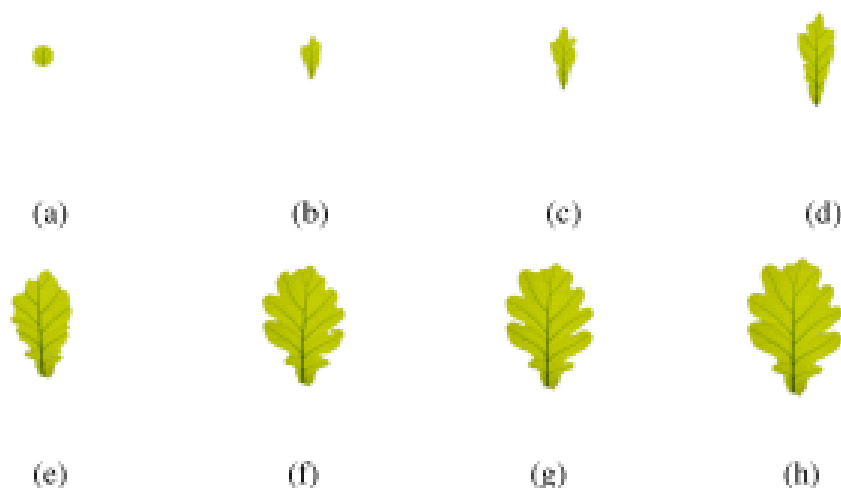
kde  $S(\Phi_0)$  je funkce znaménka definovaná jako:

$$S(\Phi_0) = \frac{\Phi_0}{\sqrt{\Phi_0^2 + (\Delta x)^2}} \quad (2.3)$$

Po reinitializaci nastává prostor pro uživatele, aby vhodnou volbou vektorového pole ovlivnil výsledný tvar listu. Vektorové pole je dvoudimenzionální (viz obr. 2.3) a udává nám velikost a směr růstu. Jeho vhodnou volbou jsme schopni ovlivnit konečný tvar. Na obr. 2.3(a) vidíme vektorové pole, které list natahuje ve všech směrech. Jak se list zvětšuje, zvyšuje se i rychlost jeho růstu. To je dáno podobou vektorového pole. Přesněji řečeno faktem, že s tím, jak se geometrický objekt zvětšuje, koresponduje s částmi vektorového pole, které se nacházejí blíže k hranám, kde je jeho intenzita větší. Na obr. 2.3(b) vidíme pole, které pomáhá vytvořit efekt ohýbání směrem k levé horní straně listu. Zvyšováním velikosti intenzity vektorového pole směrem k jeho hranám jsme schopni simulovat efekt vnějších vlivů, jako je například vítr. Třetí vektorové pole znázorněné na obr. 2.3(c) zapříčiňuje efekt smrštění jedním směrem, zatímco se list natahuje ve směru opačném. Čtvrté vektorové pole znázorněné na obr. 2.3(d) má ve středu vírový bod a velikost intenzit tohoto pole se odvíjí v závislosti na vzdálenosti od tohoto bodu. Použitím vektorového pole tohoto typu jsme schopni docílit zkrouceného listu.



**Obrázek 2.3:** Ukázka vektorového pole (převzato z: [4]).



**Obrázek 2.4:** a) Ukázka vývoje listu vyvíjejícího se z primordia (a) až do konečného tvaru (h) v závislosti na čase (převzato z:[4]).

Aplikací této práce bychom měli být schopni docílit výsledků jako na obr. 2.4.

Algoritmus nakonec používá zobecňující funkce pro směr  $x$  a směr  $y$ , což jsou směry vektorového pole, které mají za úkol spravovat růst. Funkce je spravována vhodnou volbou konfiguračních parametrů.

Vyskytují se zde ale nesrovnalosti, které v kombinaci s relativně náročnou matematikou komplikují v dostatečně krátkém čase pochopení a rekonstrukci algoritmu. Pomineme-li různé integrály, derivace a optimalizační problém, které se zde nacházejí, je používán například také algoritmus na detekci hran popsany následující funkcí:

$$g(|\Delta\mu_0(x, y)|) = \frac{1}{1 + \Delta G(x, y) * \mu_0(x, y)^p}, p \geq 1 \quad (2.4)$$

V práci není přesně řečeno, jak je funkce konkrétně využita. Její použití není na první pohled zcela zřejmé a je obtížné určit, jakou roli ve studii sehrává. Z tohoto důvodu nebude nakonec tato studie v naší práci použita.

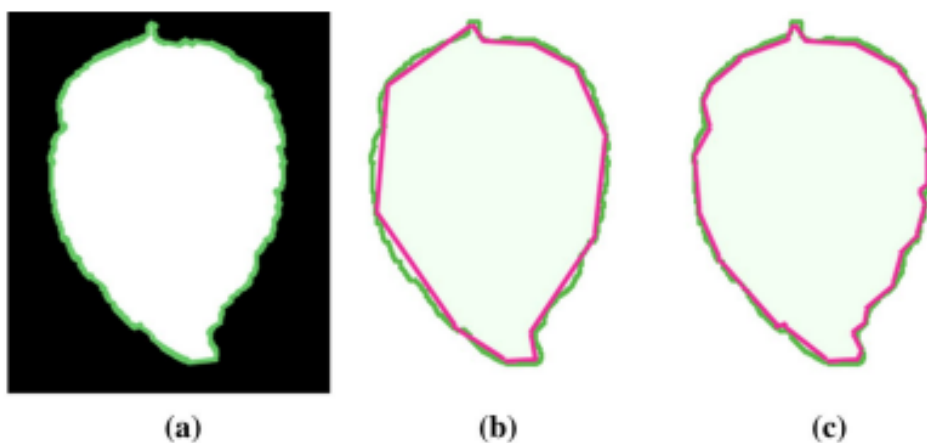
## 2.2 Procedurální modelování a vizualizace listů pomocí metody generování žilek

V naší práci budeme namísto toho vycházet hlavně ze studie Procedurální modelování a vizualizace listů pomocí metody generování žilek [1].

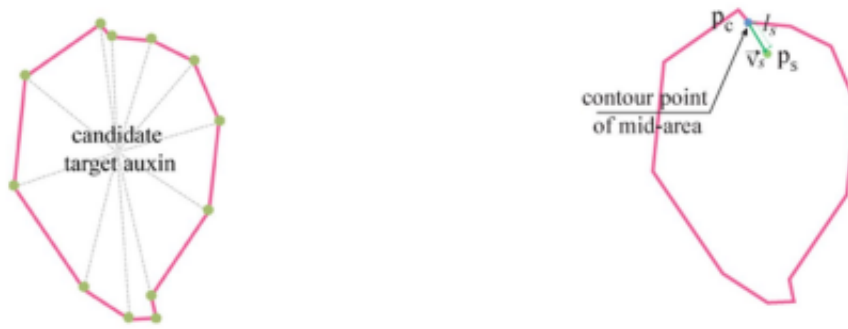
Zde popisovaná metoda začíná načtením obrázku a detekcí kontury, na jejichž základě je vygenerováno několik bodů, podle kterých jsou nejprve generovány žilky, jež jsou později použity pro generování samotného listu.

Pro růst žil je nejprve třeba vygenerovat bod, ze kterého by žíly měly růst a také do kterého bodu by měly růst. Za tímto účelem je použit obrázek s transparentním pozadím a určena jeho kontura pomocí algoritmu eight-way

search. Kontura listu je detekována pomocí Douglas-Peuckerova algoritmu [7] a následně je aproximována pomocí aproximační metody navržené ve studii od Kima [8] (viz obr. 2.5). Výsledný tvar aproximované křivky záleží na tvaru listu (viz obr.2.5 (b),(c)). Pomocí aproximované křivky jsou nalezeny kandidátní body na segmenty (viz obr.2.6 (a)). V úvalu jsou vzaty i konvexní tvary listů a aproximace je pro tyto tvary uzpůsobena. Vzor žil je určen během jejich růstu od startovního segmentu po cílový segment. Pro první segment ( $p_s$ ) je nastavena počáteční délka ( $l_s$ ) a směr ( $v_s$ ), jak je ukázáno na obr.2.6 (b).



**Obrázek 2.5:** Detekce a aproximace kontury listu za použití této metody: a) Detekovaná kontura listu, b) aproximace listu, c) detailnější aproximace listu (převzato z:[1]).



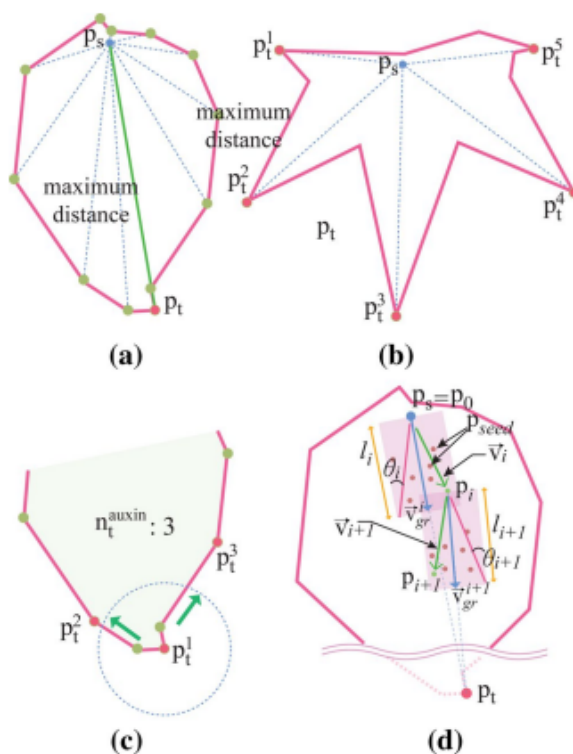
**(a)** : Výběr kandidátních bodů na segmenty z aproximované kontury.

**(b)** : Princip růstu žíly pomocí generování segmentů.

**Obrázek 2.6 :** Získané kandidátní body a růst žíly (převzato z:[1]).

Popisovaná metoda organizuje proces růstu žil ve dvou úrovních. První úroveň je růst hlavní žíly. V této úrovni je nejdříve nalezen kandidátní bod na segment, který je nejdále od počátečního segmentu a určí se za cílový (viz obr.2.7(a)) - stane se segmentem. Pokud je čepel listu konkávní, stávají se

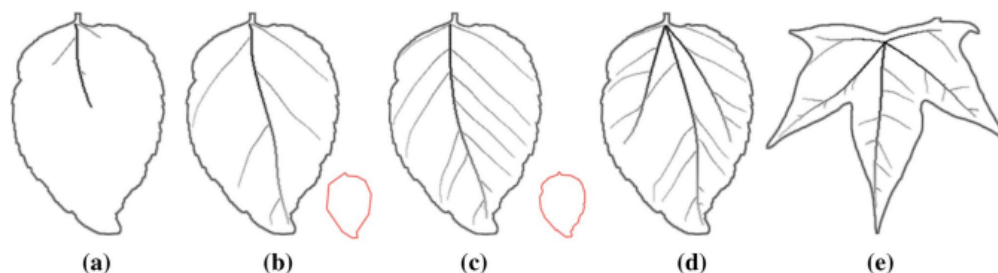
vrcholy konkávního mnohoúhelníku cílovým segmentem pro růst hlavních žil (viz obr. 2.7 (b)). Má-li být více vygenerovaných hlavních žil a jedná-li se o list konvexní, jsou zbylé segmenty vybrány z ostatních bodů kontury, které se nacházejí nejbližze za určenou prahovou zónou (viz obr. 2.7 (c)). Jakmile je nalezen cílový segment, je zahájen růst hlavní žíly (viz obr.2.7 (d)). Je spočítán vektor ve směru růstu ze startovního segmentu ( $p_s = p_0$ ) do cílového segmentu  $p_t$ .  $\delta_i$  a  $l_i$  jsou hodnoty  $\pm\mu$  určených počátečním úhlem ( $\theta$ ) a délkou ( $l$ ). Růstový prostor je vytvořen z vypočítaných  $\theta_i$  a  $l_i$  a růstového vektoru ( $v_{gr}^i$ ), který směřuje do cílového segmentu. Posléze je v růstové zóně náhodně vygenerováno  $n_{seed}$  bodů  $p_{seed}$ , kde  $n_{seed}$  je počet cílových segmentů pro hlavní žíly (segmentů bude tolik, kolik bude vygenerováno hlavních žil). Segment ( $p_i$ ) je následovně vybrán z jednoho z bodů ( $p_{seed}$ ) a je pomocí něho spočítán vektor směru růstu ( $v_i$ ). Růst žíly pokračuje, dokud nepřekone prahovou zónu. Nakonec je vytvořena Bezierova křivka, jejíž body jsou tvořeny vygenerovanými růstovými segmenty.



**Obrázek 2.7 :** Procesy detekce kandidátních bodů na segment a růst žil v závislosti na kontuře: a) detekce cílového segmentu, b) detekce cílových segmentů u konkávního tvaru listu, c) případ více cílových segmentů pro více žil d) proces růstu (převzato z: [1]).

Druhou úroveň představuje růst laterálních žil (viz obr. 2.8), jenž probíhá z hlavní žíly, jejíž segmenty jsou všechny označeny za kandidátní (startovní) body. Startovní body žil jsou procházeny proti směru hodinových ručiček a pro každý je vypočítán vektor ve směru startovního bodu nacházejícího

se na hlavní žíle do cílového kandidátního bodu nalézajícího se na kontuře. Pokud úhel mezi vypočítaným vektorem a vektorem růstu ve směru hlavní žíly překročí uživatelem definovanou hodnotu, je kandidátní startovní bod nastaven na startovní segment laterální žíly a jí náležící kandidátní cílový bod nastaven jako cílový segment laterální žíly. Kromě toho jsou v průběhu tohoto procesu laterální žíly testovány na kolize, zda se nově generovaná laterální žíla nekříží s nějakou dříve vygenerovanou laterální žilou. Pokud se s žádnou laterální žilou nekříží, je vygenerována analogickým procesem jako při růstu hlavní žíly. Výsledek generování laterálních žilek je zobrazen na obr. 2.8.



**Obrázek 2.8 :** Výsledky růstu hlavních a laterálních žil dosažených touto studií: a) Počáteční stádium vývoje, b) pozdní stádium vývoje, c) detailnější kontura listu, d) více cílových bodů pro hlavní žíly, e) konkávní tvar listu (převzato z: [1]).

Segmenty pro sublaterální a terciální žilky již nejsou určeny počátečním algoritmem, protože všechny kandidátní body již byly použity a další se musí teprve vygenerovat. Nové segmenty jsou vytvořeny na základě definovaných pravidel. Cílové kandidátní body na segment pro růst ostatních žilek listu jsou vytvořeny pomocí informací o hlavní a postranní žíle. Laterální žíla je mezi jejím startovním a koncovým segmentem rozdělena na ( $n_{div}$ ) částí a tím jsou vytvořeny kandidátní startovní body na laterální žíle. V závislosti na úhlu mezi hlavní žílou a laterálními žilami je spočten rotační úhel ( $\theta_{div}$ ), který je použit pro směr vektoru vedoucího z možných kandidátních bodů na segmenty nacházejících se na laterální žíle. Ve směru vektoru je ze startovního bodu veden paprsek, který se překříží s konturou a v tomto místě je vygenerován kandidátní cílový bod, do kterého žíla bude eventuálně růst. V případě, že by se žilka s žádnou jinou nekřížila, je vygenerována a body se stávají segmenty. Proces růstu sublaterálních a terciálních žilek však není stejný. Jelikož by tyto žilky byly tvořeny pouze počátečním a cílovým segmentem, nejevily by se dostatečně hladké. Proto je mezi každou žilku přidán do jejího středu kontrolní segment, který zajistí její hladkost.

Poslední žilka je tvořena segmenty, jež jsou náhodně rozmístěny uvnitř listu a náhodné množství segmentů je vybráno a spojeno čarou. Spojením vzniká nepravidelná síťovitá žilnatina povrchu listu (viz obr.2.9).



**Obrázek 2.9 :** Ukázka vygenerovaných bodů a vzniklých vlásovitých žilek (převzato z: [1]).

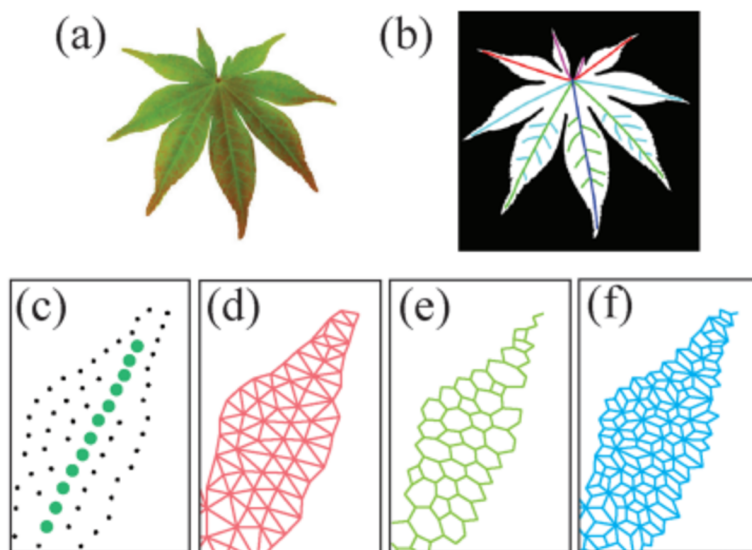
Proces tvorby listu je završen tvorbou jeho třívrstvého barevného modelu za pomoci konvolučních tabulek dělicích funkcí. Skládá se z barev, hluku a světelných map. Více informací o tvorbě barevného modelu můžeme nalézt ve studii od Kima [1].

### 2.3 Simulace morfologických změn u vysychajících listů za použití manuálně zadaných žilek

Nejdetailnější popis deformací poskytuje studie od Jeonga [6], jež se stala při tvorbě naší práce hlavní inspirací pro generování listů. V této studii je list popsán systémem hmota-pružina a zprostředkována výměna vody mezi částicemi, což zapříčiňuje změnu hmotnosti částic, zkracování a natahování pružin a deformace listu. Tato metoda nám umožňuje zkombinovat námi dosažené výsledky vygenerovaných žil s tvorbou povrchu a vrstev listu takřka bez nutnosti úprav postupu metody v této studii.

Ze všeho nejdříve je načten referenční obrázek (viz obr. 2.10 (a)) spolu s pomocným obrázkem (viz obr. 2.10 (b)), který obsahuje informace o struktuře žíly - respektive její vizualizaci. Jsou zjištěny informace z pomocného obrázku a umístěny odpovídající částice žil. Spolu s částicemi žil jsou vygenerovány částice povrchu v rovnoměrné náhodné distribuci (viz obr. 2.10 (c)), všechny částice jsou spojeny pomocí delaynovy triangulace, jež představuje spodní vrstvu listu a vygenerován voroinův diagram (viz obr. 2.10 (d)), jež představuje svrchní vrstvu listu a bude hrát hlavní roli u deformací. Vrstvy jsou následně propojeny pomocí linkové vrstvy (viz obr. 2.10 (e)), která umožňuje rozhýbat částice ve 3D prostoru tím, že spojuje částice delaynovy a voroinovy vrstvy a zprostředkovává jejich vzájemné působení.

Celá struktura je vizualizována na obr. 2.11. Uprostřed každého trojúhelníku delaynovy vrstvy (viz obr. 2.11 (a)) je ve výšce  $h$  (viz obr. 2.11 (b) a (e)) umístěna voroinova částice a spojena s částicemi příslušného trojúhelníku delaynovy vrstvy linkovou vrstvou (viz obr. 2.11 (c)). Jak je známo na

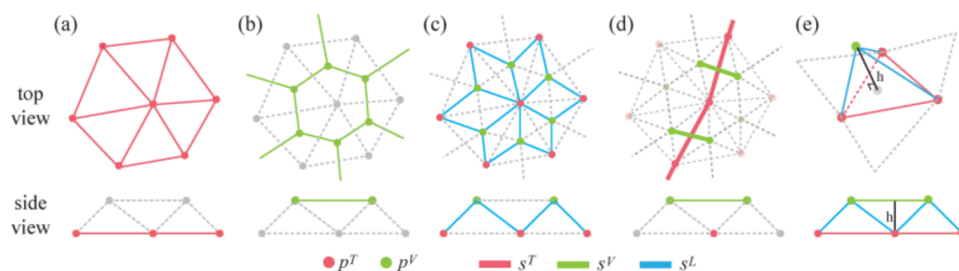


**Obrázek 2.10** : Proces modelování: a) referenční obrázek, b) pomocný obrázek s žilami, c) zelené částice žíly a černé částice povrchu, d) delaynova triangulace vzniklá z částic žil a povrchu, e) jí náležící voroinův diagram, f) linková vrstva, jež tyto dvě vrstvy spojuje (převzato z: [6]).

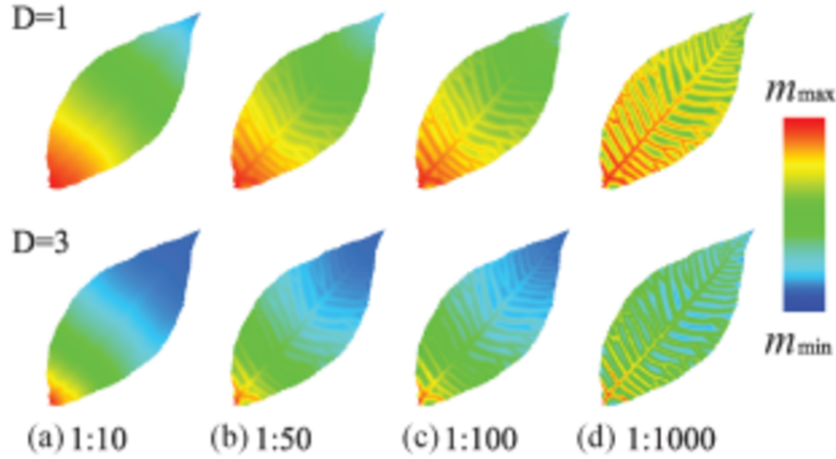
obr. 2.11(d), při pohledu seshora se vrstvy protínají. Vzdálenost voroinovy částice (viz obr. 2.11 (e)) definujeme jako nejkratší vzdálenost této částice od příslušného trojúhelníku.

Struktura tudíž disponuje dvěma druhy různých částic, a to částicemi  $p_i^T$ , které představují částice delaynovy vrstvy a částicemi  $p_i^V$ , které představují částice voroinovy vrstvy. Částice jsou zároveň propojeny třemi typy pružin, a to pružinami spojujícími částice delaynovy vrstvy  $s_j^T$ , voroinovy vrstvy  $s_j^V$  a pružinami spojujícími delaynovu vrstvu spolu s voroinovou vrstvou  $s_j^L$ .

Každá částice  $p_i$  obsahuje informace o její lokaci  $x_i$ , rychlosti  $v_i$  a navíc obsahuje hmotnost vody  $w_i$ , cytosolu  $M_i$  a celkovou hmotnost  $m_i$ . Pružina  $s_j$  spojující dvě částice  $p_a$  a  $p_b$ , uchovává informaci o její délce  $l_j$ .



**Obrázek 2.11** : Ukázka částic a pružin vrstev seshora a z boku: a) delaynova vrstva, b) voroinova vrstva, c) pružiny linkové vrstvy spojují částice voroinovy vrstvy s částicemi delaynovy vrstvy, d) pružiny voroinovy vrstvy a delaynovy vrstvy se překřížují, e) výška  $h$  udává vzdálenost voroinovy částice od příslušného trojúhelníku v delaynově vrstvě (převzato z: [6]).



**Obrázek 2.12 :** Neuniformní rozložení hmoty s hodnotami mezi  $(m_{min}, m_{max})$ . Obrázky udávají jednotlivá rozložení v závislosti na poměru váhy pružin žil a povrchu v Bellman-Fordově algoritmu a disperzním parametru  $D$ . Čím je rozdíl váhy pružin žil vůči pružinám povrchu vyšší, tím je větší hmotnost částic v okolí žil patrnější - viz obr. (d) (převzato z: [6]).

Počáteční hmota  $m_i$  částic  $p_i$  je rozmístěna neuniformně (viz obr. 2.12) podle nejkratší vzdálenosti dané částice od žil, kde vzdálenost  $d_i$  částice od částice nacházející se nejbližší řapíku listu je určena pomocí Bellman-Fordova algoritmu a pružinám spojujícím částice žil je přisouzena menší váha než částicím spojujícím povrch v poměru až 1:1000, kde tento poměr udává, jak moc bude počáteční rozložení hmoty záviset na žilách. Menší hodnota typicky zapříčiní, že hmotnost částic bude větší, čím blíže jsou k částici nacházející se nejbližší řapíku listu. V opačném případě docílíme vyšších hmotností částic hlavně v okolí žil, což je pro tuto studii žádoucí. Čím dále daná částice od žily bude, tím menší bude její hmotnost. Aby částice nacházející se dále od řapíku listu měla menší hmotnost než částice nacházející se blíže, můžeme použít například menší poměr 1:100. Interpolační parametr pro rozložení hmoty je dán následujícím vztahem:

$$\tau_i = \left(1 - \frac{d_i}{d_{max}}\right)^D \quad (2.5)$$

kde  $d_{max}$  značí vzdálenost částice nejbližší řapíku listu od částice s největší vypočtenou vzdáleností a  $D$  ovlivňuje disperzi parametru  $\tau$ . Typicky volíme  $D = \{1, 2, 3\}$ . Hmota je za využití interpolačního parametru  $\tau$  rozmístěna podle vzorce:

$$m_i^0 = m_{min}(1 - \tau_i) + m_{max}\tau_i \quad (2.6)$$

kde  $m_{min}$  a  $m_{max}$  jsou minimální a maximální uživatelem zvolené hodnoty rozsahu hmoty.

Osmotický tok vody  $L$ , jenž je vážený podle délek pružin a řeší problém neuniformní distribuce hmoty nepravidelného povrchu, je zadán následujícím vztahem:



### 2.3. Simulace morfologických změn u vysychajících listů za použití manuálně zadaných žilek

$$L(w_i) = \frac{2}{E} \sum_{j \in N_1(i)} \lambda_{ij} \frac{M_i w_j - M_j w_i}{|l_{ij}|(M_i + M_j)} \quad (2.7)$$

a kde

$$E = \sum_{j \in N_1(i)} |l_{ij}| \quad (2.8)$$

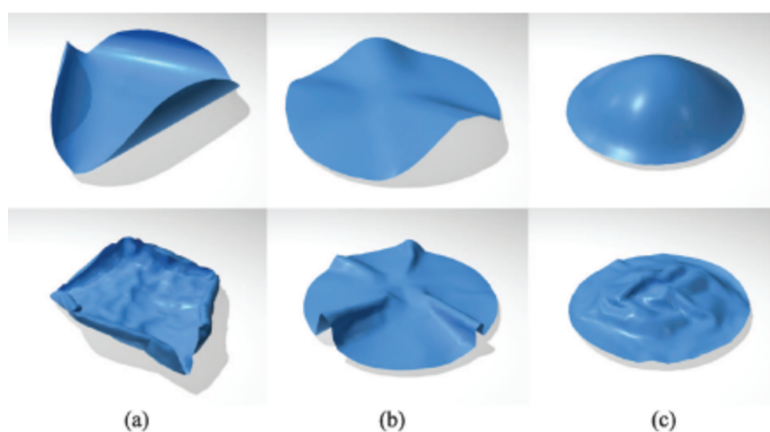
vyjadřuje součet délek všech pružin vycházejících z dané částice a  $\lambda_{ij}$  je koeficient ovlivňující rychlost difúze definovaný jako  $\lambda_{ij} = (\tau_i + \tau_j)/2$ , kde  $\tau_i$  a  $\tau_j$  jsou interpolační parametry získané při inicializaci hmoty částice  $p_i$  a  $p_j$ .

Jakmile je vypočítán difúzní koeficient vody, hmotnost vody v každé iteraci a každé částici  $p_i$  je aktualizována pomocí následujícího vztahu:

$$(I - \Delta t L)w^{n+1} = (w^n - w_{loss} \Delta t \mathbf{1}) \quad (2.9)$$

$w^{n+1}$  je aktualizovaná hmotnost vody,  $w^n$  je aktuální hmotnost vody v daném snímku  $n$ ,  $w_{loss}$  je uživatelem zvolený parametr pro rychlost vysychání všech částic,  $I$  je matice identity a  $\mathbf{1}$  je vektor, ve kterém jsou všechny elementy 1.

Ve chvíli, když se pružiny v dvouvrstvé struktuře zkracují nebo expandují, struktura se ohýbá nebo zvrásňuje, aby se minimalizovalo tahové nebo tlakové napětí. Když horní vrstva vysychá rychleji než spodní vrstva, délky pružin horní vrstvy se zkracují rychleji než ve vrstvě spodní a list se zvrásňuje směrem nahoru, jak je znázorněno na obr. 2.13(a). Tato situace v přírodě nastává nejčastěji, jelikož na horní vrstvu dopadá sluneční záření, které zapříčiňuje její rychlejší vysychání. Pokud by se délky pružin zkracovaly stejnoměrně v obou vrstvách, docílili bychom zvrásňování a ohybání v rámci roviny (viz. obr. 2.13(b) a (c)).



**Obrázek 2.13 :** Ukázka struktury hydratovaných a dehydratovaných útvarů v závislosti na difúzním koeficientu  $\lambda$ . Když je koeficient nízký, nemohou být vyschlé oblasti dostatečně rychle rehydratovány a vznikají v těchto místech nerovnosti. V opačném případě se nerovnosti rozloží po celém útvaru: a) plně hydratovaný a dehydratovaný útvar, b) s koeficientem  $\lambda = 0.0$ , c)  $\lambda = 100.0$  (převzato z: [6]).

Nová délka pružiny je vyjádřena jako:

$$l^n = l^0(1 - \sigma(1 - v_{rel})^\alpha) \quad (2.10)$$

kde

$$v_{rel} = \frac{m_a^n + m_b^n}{m_a^0 + m_b^0} \quad (2.11)$$

$m_a^0$  a  $m_b^0$  jsou hmotnosti částic, mezi kterými vede daná pružina, při inicializaci a  $l^0$  je počáteční délka pružiny při inicializaci,  $\sigma$  je uživatelem zvolený parametr odlišný pro pružiny spojující částice žil a částice povrchu ovlivňující míru srážení délky pružin a  $\alpha$  je uživatelem zvolený parametr ovlivňující změnu objemu, který při nastavení  $\alpha = 1.3$  dává lineární závislost.

Samotný pohyb částic je zprostředkován za použití metody Dynamiky založené na pozici (Position-based dynamics - PBD) [9], která iteračně mění polohy částic, dokud není pro všechny pružiny splněna podmínka  $|C/l| < \lambda_c$ , kde  $\lambda_c$  je uživatelem zvolený parametr. Spolu s polohou částic je měněna i jejich rychlost  $\Delta v_i = \Delta x_i / \Delta t$

Po PBD může nastat situace, kdy se vrstvy protnou a dojde ke kolizi. Pokud částice trojúhelníku protne voroinovou vrstvu a dojde tak ke kolizi, je pozice a rychlost této částice upravena.

Na závěr jsou pozice voroinových částic aktualizovány podle následujícího vztahu:

$$x_i^v = \frac{1}{3}(x_a^F + x_b^F + x_c^F) + h_i n \quad (2.12)$$

kde  $x_a^F$ ,  $x_b^F$  a  $x_c^F$  jsou pozice částic příslušného trojúhelníku  $F$ ,  $h_i$  je výška voroinovy částice od trojúhelníku  $F$  ještě před algoritmem PBD a  $n$  je jeho normála.

Jelikož bychom rádi zachovali výšku mezi delaynovou a voroinovou vrstvou, je výška  $h_i$  každé částice ještě dočasně uložena před aplikací algoritmu PBD pomocí následujícího vztahu:

$$h_i = (x_i^V - x_a^F) \cdot n \quad (2.13)$$

Složením předchozích kroků, od výpočtu difúzního koeficientu  $L$  počínaje, do smyčky, bychom měli být schopni dosáhnout výsledků jako na obr. 2.14.

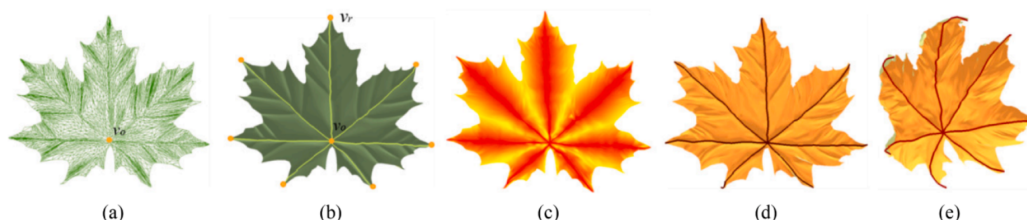


**Obrázek 2.14 :** Simulace deformace listí s různě navolenými parametry: obrázek vlevo vizualizuje deformace dosažené při vysychání obou vrstev s parametry  $\sigma_{surf} = 0.0$  a  $\sigma_{vein} = -5.0$ , obrázek uprostřed vizualizuje případ, kdy vysychá pouze voroinova vrstva s parametry  $\sigma_{surf} = 1.0$ ,  $\sigma_{vein} = 5.0$ ,  $w_{loss}^v = 2.0$ ,  $w_{loss}^T = 0.0$  a obrázek vpravo vizualizuje reálný list (převzato z: [6]).

## 2.4 Deformace listí pomocí dynamiky rostlin založené na hlavních žilách

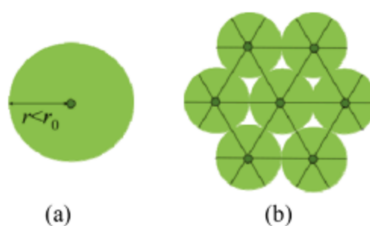
Tato nová studie z roku 2021 používá algoritmus, který je schopen rekonstrukce hlavních žil zpeřených a dlanitých listů a bezprostředně navazuje na studii pojednávající o deformacích od Jeonga [6]. Proces je rozdělen do tří částí. Nejprve je zkonstruován 3D povrch listu pomocí struktury polohran (half-

edges). Posléze je 3D model listu namapován do 2D prostoru a nakonec jsou zrekonstruovány hlavní žíly, inicializována hmota a list deformován. Při konstrukci žil jsou detekovány lokální extrémní body nacházející se na hraně listu a pomocí nich jsou za použití diskrétních geodetických cest sestrojeny hlavní žíly.



**Obrázek 2.15 :** Vývoj algoritmu: a) rekonstrukce 3D modelu listu, b) detekce cílových segmentů hlavních žil, c) neuniformní rozmístění hmoty v závislosti na zrekonstruovaných žilách, d) deformace vrásek, e) zvlnění a kroucení (převzato z: [5]).

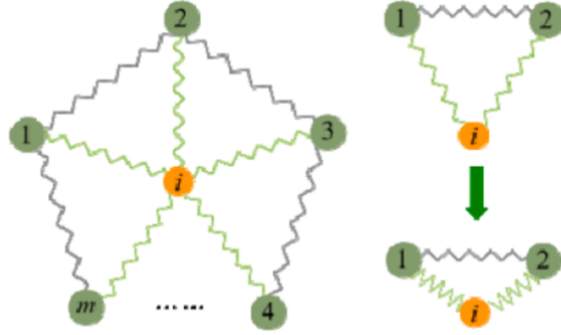
Poté, co je zrekonstruován 3D model listu (viz. obr.2.15(a)) a sestrojeny žíly (viz obr.2.15(b)), je uvnitř listu neuniformně rozložena hmota (viz obr.2.15(c)), jež je později využita pro efekty deformací a kroucení (viz obr.2.15(d) a (e)).



**Obrázek 2.16 :** Ukázka buněk a jejich struktury: a) buňka listu o počátečním poloměru  $r_0$ , který se časem mění -  $r$ , b) ukázka systému hmota-pružina pro více buněk (převzato z: [5]).

Výplň listu je tvořena buňkami, přičemž každá buňka je vyplněna cytosolem a vodou.  $r_0$  a  $r$  (viz obr. 2.16) pak reprezentují poloměr buňky ve stádiu, kdy je buňka plně hydratována a poloměr buňky poté, co je dehydratována a sražena.

Aby byly lépe simulovány síly působící mezi buňkami, je zaveden tzv. systém hmota-pružina (viz obr. 2.17). Každá pružina spojuje částice (celkem tři pružiny na každý trojúhelník), jež spolu tvoří trojúhelník (pravděpodobně inspirováno principem delaynovy triangulace). Hmota částice  $m_i$  je rovna součtu hmotnosti vody  $w_i$  a cytosolu  $b_i$ . Jelikož musí každá částice obsahovat jiné množství hmoty v závislosti na vzdálenosti od žíly, protože chceme docílit reálného a neuniformního rozdělení, jsou hmotnosti částic lineárně interpolovány v závislosti na vzdálenosti dané částice od nejbližší žíly následovně:



**Obrázek 2.17** : Systém hmota-pružina: obrázek vlevo symbolizuje rozdělení částic do trojúhelníků, obrázek vpravo ukazuje princip deformace, kdy se zkracují pružiny mezi částicemi, a tím je ovlivňována pozdější deformace (převzato z: [5]).

$$m_i^0 = m_{min} \left( 1 - \left( 1 - \frac{gd_i}{gd_{max}} \right)^t \right) + m_{max} \left( 1 - \frac{gd_i}{gd_{max}} \right)^t \quad (2.14)$$

kde  $t \in (0, 1)$  je koeficient distribuce hmoty. Pokud  $t = 1$ , počáteční hmota buněk klesá lineárně se vzdáleností od žíly. Pokud  $t \neq 1$ , klesá nelienárně.  $m_{min}$  a  $m_{max}$  jsou minimální a maximální hodnoty hmoty buněk,  $gd_i$  je vzdálenost částice od částice nacházející se nejbližše řapíku listu a  $gd_{max}$  je vzdálenost částice nacházející se nejdále od řapíku listu.

Jak list stárne, vysychá, což je modelováno parametrem  $w_{loss}$ , a probíhá osmotický tok. Hmotnost vody každé částice je v každé iteraci smyčky aktualizována podle následujícího vztahu:

$$w^{n+1} = (w^n - w_{loss}\Delta t)(1 - \Delta t L_i^n)^{-1} \quad (2.15)$$

kde  $w_{loss} = THK$ ,  $T$  je teplota,  $H$  udává vlhkost,  $K$  je řídicí parametr, který závisí na druhu listu a  $L$  je Laplaceův operátor pro osmózu definovaný takto:

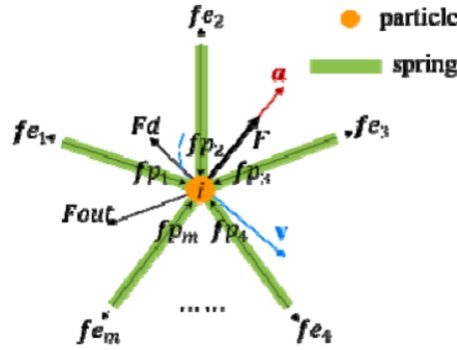
$$L_i = \frac{2}{E} \lambda \frac{b_i w_j^n - b_j w_i^n}{l_{ij}(m_i^n + m_j^n)}, i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (2.16)$$

$E$  získáme jako součet všech délek pružin vycházejících z dané částice:

$$E = \sum l_{ij} \quad (2.17)$$

a  $n$  značí celkový počet částic,  $m$  je počet částic, do kterých vedou pružiny z dané částice,  $\lambda$  je koeficient ovlivňující rychlost difúze vody,  $l_{ij}$  udává délku pružiny,  $w_i^n$  udává hmotnost vody částice,  $b_i$  hmotnost cytosolu a  $m_i^n$  její celkovou hmotnost.

Abychom dosáhli realistických výsledků, splňuje pohyb částice  $p_i$  zákony buněčné mechaniky, které jsou určeny faktory zahrnujícími hmotu  $m_i$ , prostorové souřadnice  $x_i$ , a výsledné síly  $F_i$  (viz. obr. 2.18). Kromě definování



**Obrázek 2.18 :** Obrázek popisující vypětí a pohyb částice  $p_i$ . (převzato z: [5]).

pružné síly  $F_{e_i}$  a tlumicí síly  $F_{d_i}$  pro udržení stability systému definujeme také permeační expanzní sílu  $F_{p_i}$  buněk a vnější síly  $F_{o_i}$ , které jsou proměnné v čase a působí na částice  $p_i$ . Výslednou sílu  $F_i$  působící na částice  $p_i$  můžeme získat podle vztahu:

$$F_i = F_{e_i} + F_{d_i} + F_{p_i} + F_{o_i}, i = 1, 2, \dots, n \quad (2.18)$$

kde sílu  $F_{e_i}$  můžeme spočítat jako:

$$F_{e_i} = k_s \sum (|x_{ij}| \cdot l_{ij}) x_{ij} / |x_{ij}|, i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (2.19)$$

kde  $F_{e_i}$  značí výslednou sílu působící na částici  $p_i$  a kde  $k_s$  udává koeficient pružnosti.  $l_{ij}$  značí počáteční délku pružiny,  $|x_{ij}|$  pak délku pružiny v daný moment. Tlumicí síla  $F_{d_i}$  umožňuje vyhnout se nestabilitě způsobené nadměrnými oscilacemi vyplývajícími z pohybu částice a je popsána vztahem:

$$F_{d_i} = k_d \sum (v_j \cdot v_i), i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (2.20)$$

kde  $k_d$  je parametr ovlivňující velikost tlumicí síly a  $v_j$  a  $v_i$  jsou rychlosti částic  $p_j$  a  $p_i$ .

Kapalina uvnitř buňky je obalena buněčnou stěnou a plazmatickou membránou, která vytváří tlak. Rozdíl mezi tímto tlakem a vnějším standardním atmosférickým tlakem je definován jako bobtnací síla. Síla bobtnání buněk  $F_{p_i}$  působící na částice  $p_i$  je definována jako:

$$F_{p_i} = k_p \sum a_{ij} \Delta p_i x_{ij} / |x_{ij}|, i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (2.21)$$

kde  $k_p$  je koeficient bobtnání a  $a_{ij}$  značí plochu  $j$ -tého trojúhelníku mezi  $m$  trojúhelníky vzniklými spojením částice  $p_i$  s  $m$  částicemi v okolí, do kterých z této částice vede pružina a  $\Delta P_i$  je rozdíl mezi vnitřním a vnějším tlakem částice  $p_i$  v daném okamžiku.

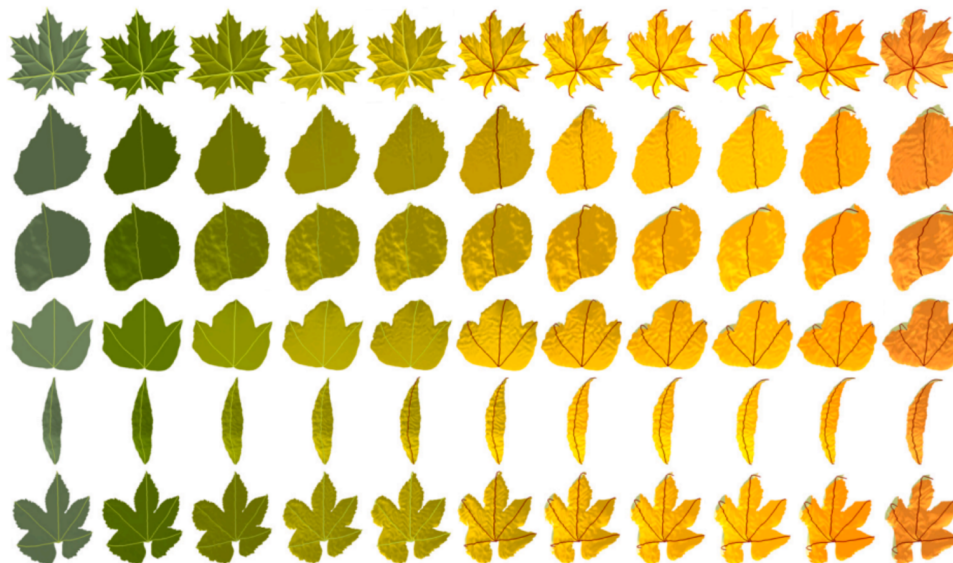
Po nakonfigurování správných parametrů, jako je difúzní koeficient  $\lambda$ , minimální a maximální množství rozložené hmoty  $m_{min}$  a  $m_{max}$ , atd..., můžeme aktualizovat rychlosti a pozice částic podle následujícího vztahu:

$$v_i^{n+1} = v_i^n + \Delta t F \left( \frac{x_i^{n+1} + x_i^n}{2} \right) / m_i^n, i = 1, 2, \dots, n \quad (2.22)$$

$$x_i^{n+1} = x_i^n + \frac{\Delta t}{2} (p_i^{n+1} + p_i^n), i = 1, 2, \dots, n \quad (2.23)$$

kde  $m_i^n$  udává hmotu částice  $p_i$  v daný čas.

Aplikováním předešlých kroků bychom měli být schopni simulovat podobné deformace jako na obr. 2.19.



**Obrázek 2.19 :** Ukázka procesu postupné deformace listů s využitím této metody (převzato z: [5]).

Nevýhodou této studie je, že pro deformace zvažuje pouze hlavní žíly, neposkytuje podrobný popis některých částí, který by byl třeba hlavně v úvodní části, jako je konstrukce žil, které jsou později využity, převod modelu z 2D do 3D, jak je přesně zkonstruován povrch a navíc nedosahuje tak pohledných výsledků jako předchozí studie o deformacích od Jeonga [6]. Přesto jsou některé části studie užitečné a naše práce se jimi inspiruje.





## Kapitola 3

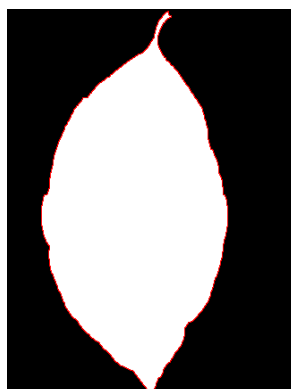
### Návrh řešení a realizace

V této kapitole se zaměříme na detailní postup řešení. Nejdříve se podíváme na načítání segmentů pro generování žil a konfigurační soubor, posléze se budeme věnovat návrhu implementace samotných žil a na závěr popíšeme proces deformací. Zkombinujeme popsané metody pro generování listů a procedurální deformace, drobně je případně upravíme a ukážeme průběžné výsledky. Pro každý průběžný výsledek vysvětlíme, co má společného s předchozími metodami, popíšeme jeho princip a poskytneme jeho vizualizaci.

#### 3.1 Generování segmentů a konfigurační soubor

Pro detekci počátečních segmentů určujících růst žil byla užitá studie Procedurální modelování a vizualizace listů. Pomocí metody generování žilek [1] využívající Eight-way search a Douglas-Peuckerovův algoritmus, detekující z daného referenčního obrázku konturu listu, se aproximují a získávají kandidátní body na segmenty, jež jsou použity pro růst žil. Toto řešení však nemusí fungovat ideálně pro každý typ listu, a tak načítáme obrázek s černým pozadím a převádíme ho na černobílý. Pro detekci kontury jsme si vypůjčili Eight-way search algoritmus (viz obr.3.1) od Martina Pažouta [2], jinak jsme v této části přišli s vlastním řešením, kdy načítání segmentů hlavních žil řešíme modelovým zápisem parametrů do konfiguračního souboru. Abychom dosáhli námi vytouženého listu, stačí navolit správně parametry v tomto souboru. Tímto způsobem zajistíme přesné vygenerování požadovaných segmentů daného listu a ovlivníme následný růst žil.

Konfigurační soubor neobsahuje pouze údaje o startovních a koncových segmentech hlavních žil, ale udává také parametry jako je lokální a globální úhel, pod kterým žíly rostou, tloušťku žil, spravuje úhly, pod kterými žíly rostou, jejich rozestupy, střídavost žil, barvu listu, obsahuje konkrétní body kontury, které využíváme pro triangulaci povrchu a další. Kromě možnosti definování parametrů se za běhu programu generují další náhodné faktory ovlivňující růst žil. Výhoda spočívá v tom, že v konfiguračním souboru můžeme nadefinovat nespočet různých modelů listů a díky vestavěné náhodnosti programu generování listů, přestože bude náš konfigurační soubor obsahovat jen malé množství modelů, nebude žádný list stejný. Konfigurační soubor může sloužit jako velká databáze dat, ze které je při generování listů jednoduše vybíráno



**Obrázek 3.1 :** Znáznornění nalezené kontury obrázku listu s černým pozadím pomocí červené barvy.

pomocí náhodné selekce listu z dat. Dosavadní implementace umožňuje selekci dat z konfiguračního souboru na základě uvedeného parametru při spuštění programu.

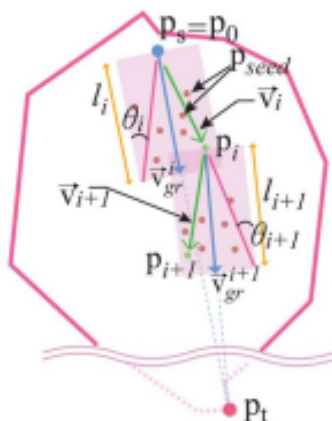
## 3.2 Růst hlavní žíly

Podobně jako ve studii od Kima [1] popisujeme modelování listů založené na generování žilek a růst hlavní žíly probíhá ze startovního segmentu do cílového segmentu. Jedinou odlišností je, že segmenty v této práci nejsou generovány algoritmem, nýbrž nakonfigurovány uživatelem. Následný vývoj růstu však odpovídá již výše popsané metodě založené na generování žilek až na několik drobností.

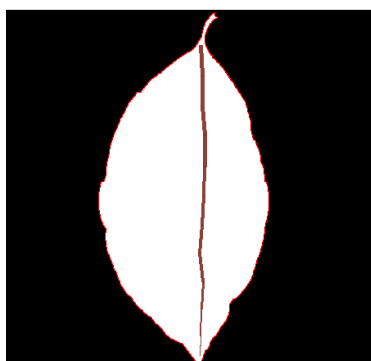
Růst hlavní žíly je ovlivňován několika faktory. Nejdříve jsou z konfiguračního souboru načteny parametry v podobě počátečního a koncového segmentu, úhlů, prahové zóny, tloušťky a barvy v závislosti na volbě typu listu uloženého v konfiguračním souboru pomocí přepínače, kterým vybíráme konkrétní model listu z databáze. Jednotlivé parametry ovlivňují růst hlavní žíly.

Podobně jako je v metodě pro generování žilek od Kima [1] spočítán pro každý segment vektor  $v_{gr}$ , což je vektor směru růstu z počátečního segmentu do cílového segmentu, je v naší práci v každém segmentu počítán úhel a vzdálenost do cílového segmentu z momentálního bodu  $p_i$ . Rozdíl však nastává v generování dalších segmentů hlavní žíly. Namísto generování několika náhodných bodů v růstovém prostoru (viz obr.3.2), který je na obrázku vyznačen jako růžová zóna, jež je definována délkou  $l_i$  a úhlem  $\theta_i$  a kde je počet náhodně vygenerovaných segmentů v určitém úhlu  $\pm\mu$  roven počtu cílových segmentů, je v této práci generován pouze jeden jediný bod, který je přidán jako segment. Vygenerování bodu je náhodné a nakonfigurovatelné pomocí parametrů. První parametr  $\omega$  udává globální konstantní úhel, což si lze představit jako vektor, který v každé fázi tlačí potenciálně generovaný bod na levou či pravou stranu listu podle toho, zda je úhel záporný či kladný. Dále definujeme druhý parametr  $\kappa$  udávající lokální v každé iteraci proměnný

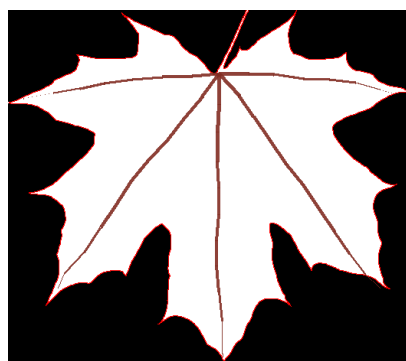
úhel, který slouží k náhodnému vychýlení směru generovaného segmentu. Výsledná pozice bodu je dána součtem úhlů  $\omega$  a úhlu, který svírá vektor  $\vec{v}_{gr}$  s vertikální osou. Konečný úhel, pod kterým je segment vygenerován je ještě sečten s lokálním úhlem  $\kappa$ . Třetí parametr  $s$  nám vektor natáhne do délky o definovanou hodnotu. Dalším rozdílem je, že vykreslování žíly probíhá až do konce. Při překročení prahové hranice se žíla jen dostatečně ztenčí, což má za následek, že v těchto místech nepovedou žádné jiné žíly. Výsledek růstu hlavních žil je vizualizován na obr. 3.3.



**Obrázek 3.2 :** Především obrázek znázorňující průběh růstu žíly (převzato z: [1]).



(a):



(b):

**Obrázek 3.3 :** Vygenerované hlavní žíly: a) hlavní žíla konvexního listu, hlavní žíly konkávního listu.

### 3.3 Laterální a sublaterální žíly

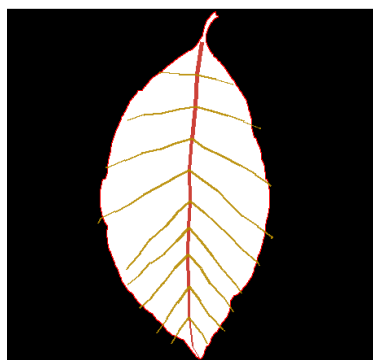
Pro laterální žíly se, na rozdíl od studie od Kima, v tomto kroku cílové kandidátní segmenty nejprve vygenerují. Pomocí parametrů v konfiguračním souboru můžeme upravovat jak množství žil a tedy i cílových segmentů, kterých chceme dosáhnout, tak i náhodný faktor ovlivňující úhel, pod kterým mají jednotlivé žíly růst. Po určení startovních a cílových bodů probíhá

proces růstu žil téměř identicky jako u růstu hlavní žíly s tím rozdílem, že růst již není ovlivněn globálním úhlem  $\omega$ . Laterální žíly se navíc testují na kolize. Vhodně zadanými parametry v konfiguračním souboru bychom sice byli schopni dosáhnout takových tvarů žilek, aby se přes sebe nekřížily, nicméně rozsah parametrů by tím byl značně omezen a míra náhodnosti by musela být dostatečně malá tak, aby se žilky nekroutily takovým způsobem, že by se protínaly. Testování kolizí se provádí způsobem, kdy je vzat startovní segment kandidátní žíly  $p_s$  a koncový segment kandidátní žíly  $p_k$ . Mezi těmito body je vytvořena imaginární úsečka, která se testuje na kolize se všemi ostatními imaginárními úsečkami vytvořenými ostatními dvojicemi počátečních a koncových segmentů žil

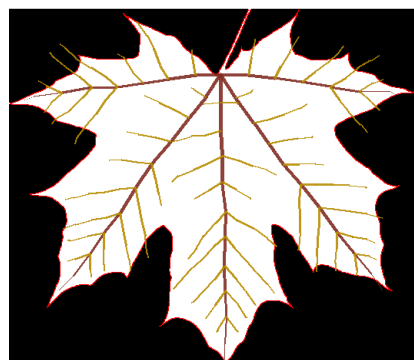
$$\{(p_{(0)s}, p_{(0)k}), (p_{(1)s}, p_{(1)k}), \dots, (p_{(n)s}, p_{(n)k})\}, \quad (3.1)$$

kde  $n$  je počet všech dosavadních existujících segmentů žil.

Výsledek růstu laterálních žil je vizualizován na obr. 3.4.



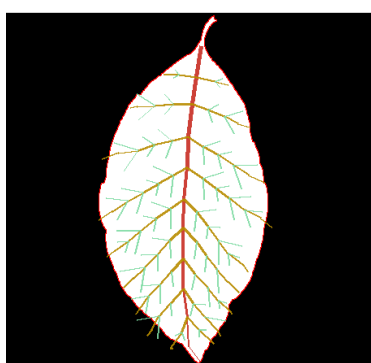
**(a)** : Hlavní a laterální žíly konvexního listu.



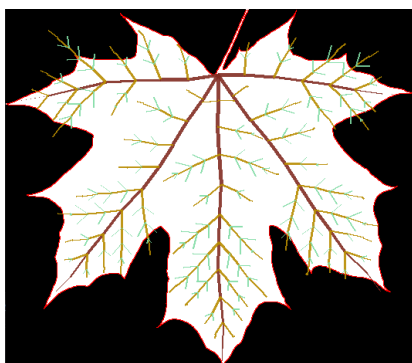
**(b)** : Hlavní a laterální žíly konkávního listu.

**Obrázek 3.4** : Vygenerované hlavní, laterální a sublaterální žíly.

Startovní segmenty sublaterálních žilek se nacházejí na laterálních žilách a jsou generovány totožně jako u laterálních žilek. Počet a směr sublaterálních žilek je také ovlivňován konfiguračními parametry. Narozdíl od metody pro generování žilek od Kima [1], kde růst sublaterálních a terciálních žilek udává startovní, cílový a mezi nimi vygenerovaný kontrolní segment, řídí v této práci růst žilek pouze startovní a cílový bod. Jelikož jsme nestihli implementovat bezierovu křivku a kontrolní segment, nabývají sublaterální žilky pravidelných, ne příliš přirozených tvarů (viz obr. 3.5).



(a):



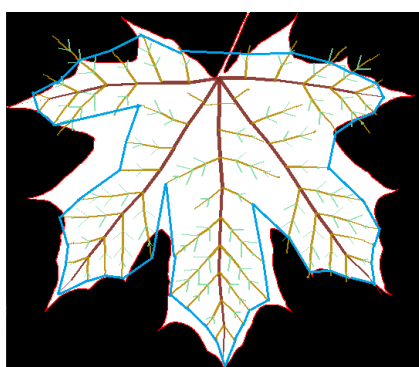
(b):

**Obrázek 3.5 :** Vygenerované hlavní, laterální a sublaterální žíly: a) hlavní a laterální žíly konvexního listu, b) hlavní a laterální žíly konkávního listu.

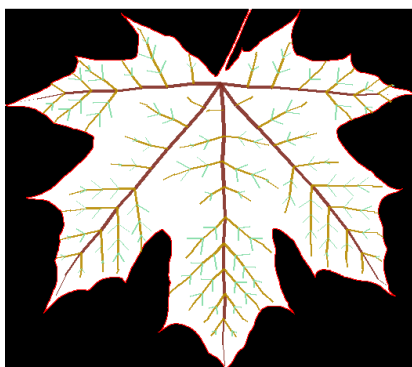
### 3.4 Detekce překročení žilek za oblast kontury

Původně bylo zamýšleno, že výsledný tvar listu by mohl být vytvořen pospojováním konců bodů hlavních a laterálních žil. Jak je ale ukázáno na obr.3.6 (a), výsledná kontura například u javoru, která by vznikla takovýmto pospojováním, by se jen těžko přiblížila skutečné kontuře javoru.

Právě proto nakonec v této práci používáme konturu jako mezník, který žíly nesmějí překročit (viz obr.3.6 (b)). V úvahu přicházely dva různé přístupy. První přístup by znamenal pro každý nově vznikající úsek dané žíly ověřovat, zda vykreslením tohoto úseku nepřekročíme hranici kontury. Žílu bychom od daného úseku tedy nevykreslovali. Druhý přístup by mohl být takový, že bychom postupně proiterovali celý obrázek a žíly bychom, v případě, že překročily konturu, vymazali.

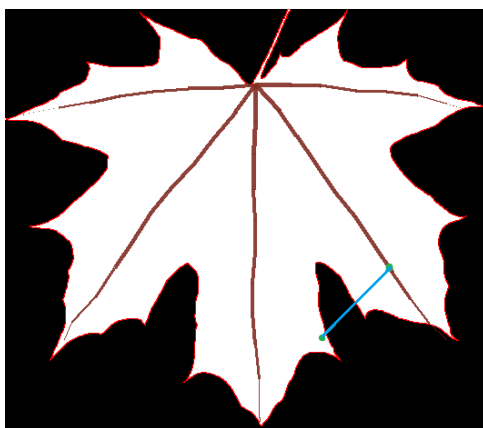


(a):



(b):

**Obrázek 3.6 :** Listy podle kontury: a) Nejlepší možná dosažitelná kontura javoru vzniklá pospojováním bodů některých laterálních a hlavních žil, b) ponechaná kontura javoru zbavená o překračující žíly.



**Obrázek 3.7 :** Spojení počátečního a vygenerovaného cílového segmentu nalézajících se uvnitř kontury listu, které vystupuje z kontury

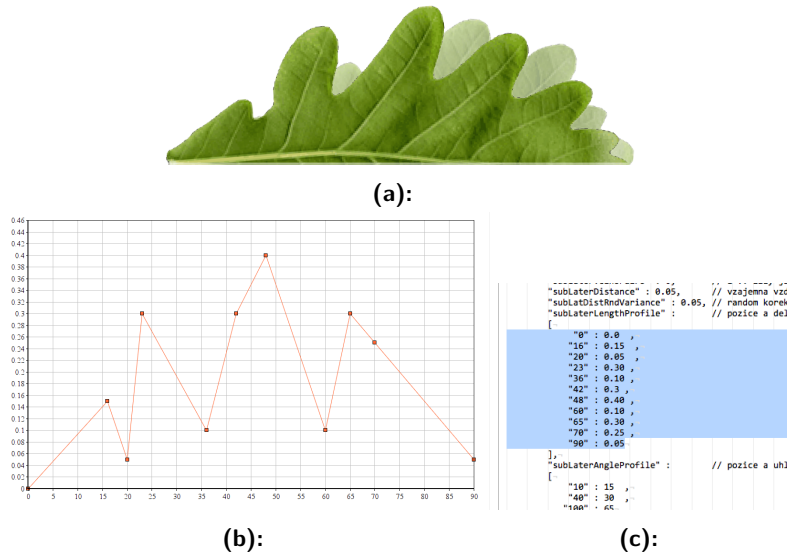
Jelikož pracujeme pouze s obrázky s černým pozadím, které jsou převedeny na černobílé, jeví se druhý přístup jako poměrně triviální a výpočetně i časově málo náročný. Princip spočívá pouze v tom, že pro každý cílový segment dané žíly určíme, zda odpovídající pixel pro tento segment má černou barvu nebo barvu kontury. V takovém případě víme, že se segment nachází vně kontury listu a žílu od daného úseku již dále nevykreslujeme. Tento přístup by se však potýkal s problémy (viz obr. 3.7). Jak je znázorněno na obrázku, v tomto případě by se cílový segment nalezal uvnitř kontury a žilka by se vykreslila. Druhý přístup by navíc sice většinou zamazal všechny žíly vystupující mimo konturu, nicméně všechny segmenty by v programu zůstaly. To by mohlo způsobovat problémy při pozdějších fázích této práce, jako je například stárnutí.

U prvního použitého přístupu je nutné procházet úsek od počátečního segmentu až po cílový segment a detekovat pro každý potenciálně vykreslený pixel, zda nebude kolidovat s hranou kontury. Tento přístup je sice složitější než druhý, nicméně není výpočetně náročný.

### 3.5 Generování profilu podle konfiguračního souboru

Jak jsme již uváděli, pro tvorbu listu musíme v konfiguračním souboru vymodelovat jeho patřičný profil. Toho dosáhneme nastavením profilových parametrů v konfiguračním souboru. Na obr. 3.8 (a) můžeme vidět model listu dubu, který se pokusíme vymodelovat zadáním odpovídajících parametrů. Levá strana profilu listu je znázorněna sytější barvou, zatímco pravá strana profilu listu, která se na obrázku nachází za levou stranou profilu, je zvýrazněna jako méně sytá. Můžeme si tedy všimnout, že list není symetrický. Na obr.3.8 (b) vidíme graf, který prezentuje závislost délky generovaných žilek na procentech délky listu, kdy procenta udávají poměrnou část listu vzhledem k jeho celkové délce od řapíku až ke konci. Menší nevýhodou je, že proces generování

předpokládá symetričnost a délky žilek na obou stranách budou podobné. Tomu ale lze předejít nakonfigurováním správného místa, ze kterého bude žilka růst a jejím správným profilem. Obr. 3.8 (c) pak vizualizuje ukázkou z konfiguračního souboru pro tvorbu profilu. Číslo nalevo v označeném textu v uvozovkách udává procentuální poměr pozice žilky vzhledem k celkové délce listu v procentech a číslo napravo udává délku, jaké mají žilky v této části dorůst.

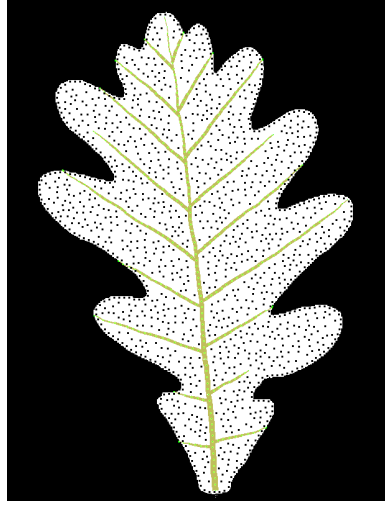


**Obrázek 3.8 :** Profil listu: a) list buku složený do jedné poloroviny, b) ukázka grafu matematické formulace vymodelování, c) ukázka profilu laterálních žil v konfiguračním souboru.

### 3.6 Struktura listu a inicializace hmoty

V této fázi máme již vygenerované hlavní, laterální a sublaterální žilky a jejich segmenty. Abychom vytvořili povrch listu, musíme podle již zmiňované metody zaměřené na deformaci listů pomocí žilek od Jeonga [6] nejdříve s náhodnou rovnoměrnou distribucí vygenerovat částice povrchu uvnitř listu (pro připomenutí viz obr. 2.10). Vizualizace rozložení částic je na obr.3.10. Jak si můžeme všimnout, kontura je rovnoměrně pokryta černými částicemi povrchu, které jsou ještě před samotným umístováním částic povrchu vytvořeny pomocí jednoduchého algoritmu podobného Eight-way search, který postupuje po kontuře a v její blízkosti vytváří v pravidelných intervalech částice. Kdybychom částice kontury negenerovali, povrch listu na jeho okrajích by se jevil nepřirozený a trhaný. Po rozmístění částic kontury následuje rozmístění částic povrchu uvnitř listu. Částice povrchu je vygenerována, pokud se v jejím okolí daném kružnicí o poloměru  $r$ , kde  $r$  je pevná hodnota zadaná pro tento program, aby byly výsledky co nejlepší, nenachází žádná jiná částice a zároveň se daná generovaná částice nachází uvnitř listu.

Jakmile jsou rozmístěny všechny částice, můžeme s jejich pomocí vytvořit



**Obrázek 3.9 :** Rozmístění částic žil (zelená) a částic povrchu (černá).

dvouvrstvou strukturu, kde spodní vrstva této struktury je tvořena z delaynovy triangulace (viz obr. 3.10(a)) a svrchní vrstva z voroinových diagramů (viz obr. 3.10(b)). Obě vrstvy jsou propojeny pomocí linkové vrstvy (viz obr. 3.10(c)). Dohromady spolu tvoří strukturu, která je vizualizována na obr.3.10(d)). Struktura se tudíž skládá ze dvou různých druhů částic  $p_i^T$  a  $p_i^V$  a třech druhů pružin  $s_j^T$ ,  $s_j^V$  a  $s_j^L$ . Každá částice si uchovává polohu  $x_i$ , rychlost  $v_i$ , hmotnost cytosolu  $M_i$ , hmotnost vody  $w_i$  a celkovou hmotnost  $m_i$ . Pružiny si uchovávají jejich délku při inicializaci  $l_0$  a momentální délku  $l_i$ .

Po vytvoření vrstev potřebujeme vhodně rozmístit hmotu. Studie od Jeonga [6] pro procedurální deformace listů za použití žilek přichází s rozmístěním hmoty v závislosti na žilkách listu za použití Bellman-Fordova algoritmu. Aby rozmístění hmoty bylo korektní, musí spolu každé dvě sousední částice žil tvořit trojúhelník (viz obr. 3.11) a pružinám žil musí být přisouzena jiná váha než pružinám spojujícím částice povrchu. V této práci jsme pružinám hlavních žil přiřadili váhu 1, pružinám laterálních žil váhu 50, sublaterálních žil váhu 200 a pružinám spojující částice povrchu váhu 1000. Volba takovýchto vah zapřičiňuje rozumné neuniformní rozdělení hmoty, kde částice nacházející se blíže k žilám nesou větší množství hmoty a zároveň s tím, čím je daná částice dále od řápíku listu, její hmota klesá.

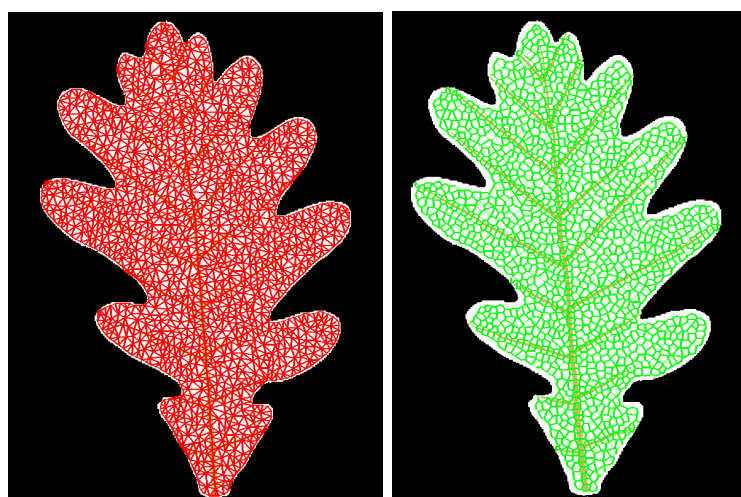
Podle Jeonga je počáteční hmota každé částice dána následujícím vztahem:

$$m_i^0 = m_{min}(1 - \tau_i) + m_{max}\tau_i \quad (3.2)$$

kde  $m_{min}$  a  $m_{max}$  jsou nejmenší a největší možné množství hmoty částice a kde  $\tau$  je interpolační parametr závisející na vzdálenosti dané částice a částice s největší vzdáleností vypočtených pomocí Bellman-Fordova algoritmu zadaného následovně:

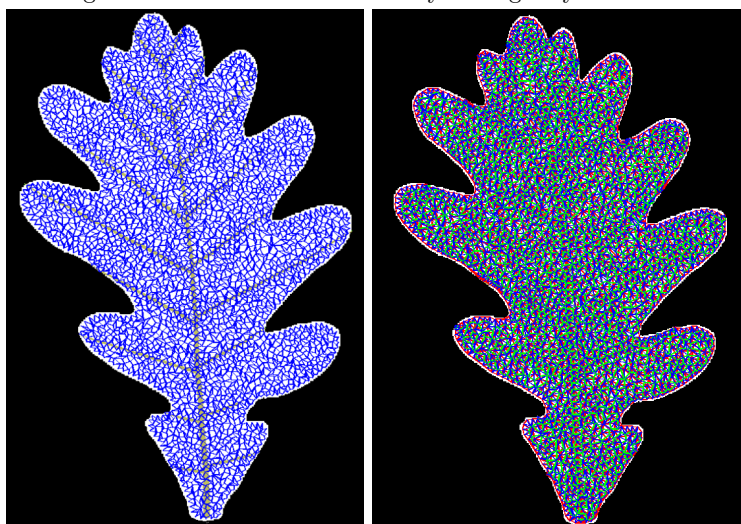
$$\tau_i = \left(1 - \frac{d_i}{d_{max}}\right)^D \quad (3.3)$$





(a) : Vrstva tvořená delaynovou triangulací.

(b) : Vrstva tvořená voroinovými diagramy.



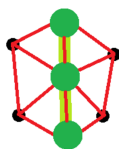
(c) : Linková vrstva propojující delaynovou a voroinovou vrstvu.

(d) : Celá struktura listu je tvořena delaynovou, voroinovou a propojující linkovou vrstvou.

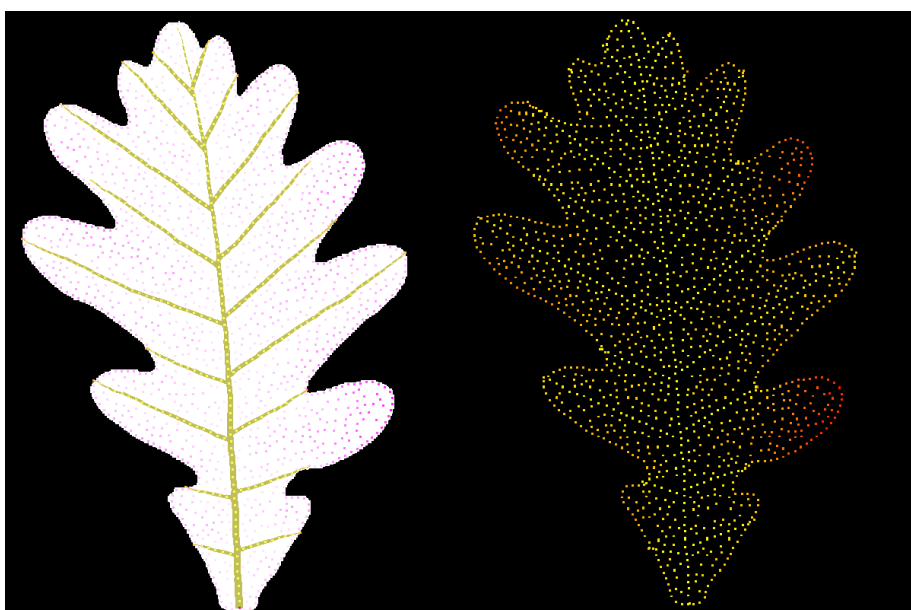
**Obrázek 3.10** : Dvouvrstvá struktura vizualizovaná na obrázku dubu.

kde  $d_{max}$  značí vzdálenost částice nejbližší řápíku listu od částice s největší vypočtenou vzdáleností,  $d_i$  je vzdálenost dané částice  $i$  a  $D$  ovlivňuje disperzi parametru  $\tau$ .

V místech, kde se nenechází žíly, je díky neuniformní distribuci hmoty závisející na vzdálenosti částice od počátku listu a od jeho žil hmotnost částic mnohem menší než v jejich okolí (viz obr.3.12).



**Obrázek 3.11 :** Částice žil spolu tvoří trojúhelník, jinak by počáteční distribuce hmoty nebyla správná a neprobíhal by správný osmotický tok vody.



**Obrázek 3.12 :** Dvě různé vizualizace rozložení hmoty: na obrázku vlevo je hmotnost částic vizualizována sytostí barvy. Čím je barva sytější, tím menší má daná částice hmotnost. Na obrázku vpravo je vizualizována hmotnost částic povrchu a i žil. Čím je barva částice blíže červené, tím menší má hmotnost. Částice žil mají typicky velkou hmotnost, jelikož jsou spojeny pružinami s malými hmotnostmi.

### 3.7 Osmóza vody a deformace

Jakmile je vytvořena struktura listu a rozmístěna hmota, můžeme simulovat vysychání, deformace a osmózu, která probíhá přesně stejným způsobem jako u předchozích metod pojednávajících o deformacích [5] [6]. Podle metody pro deformace listu za využití žilek od Jeonga pro výpočet osmózy potřebujeme vypočítat difúzní koeficient  $L$  podle vztahu:

$$L(w_i) = \frac{2}{E} \sum_{j \in N_1(i)} \lambda_{ij} \frac{M_i w_j - M_j w_i}{|l_{ij}|(M_i + M_j)} \quad (3.4)$$

kde

$$E = \sum_{j \in N_1(i)} |l_{ij}| \quad (3.5)$$

Z metody pro deformace listů za použití hlavních žilek od Suna převezmeme

vzorec na výpočet osmózy vody a vysychání:

$$w^{n+1} = (w^n - w_{loss}\Delta t)(1 - \Delta t L_i^n)^{-1} \quad (3.6)$$

Rozšířením této rovnice o uživatelsky zadaný parametr  $\lambda$  pro ještě rychlejší difúzi vody dostáváme rovnici:

$$w^{n+1} = (w^n - w_{loss}\Delta t)(1 - \Delta t \lambda L_i^n)^{-1} \quad (3.7)$$

Celkovou hmotnost částice  $p_i$  aktualizujeme podle vztahu:

$$m_i^n = M_i + w_i^n \quad (3.8)$$

Následováním kroků metody pro deformace listů za využití žilek od Jeonga víme, že na základě aktualizované hmotnosti částic se změnila koncentrace vody vůči cytosolu a to zapříčiní zkracování pružin  $l$ . Pro výpočet nové délky pružiny potřebujeme znát objem částice:

$$v_{rel} = \frac{m_a^n + m_b^n}{m_a^0 + m_b^0} \quad (3.9)$$

a následně můžeme aktualizovat délku všech pružin pomocí vztahu:

$$l_i^n = l^0(1 - \sigma(1 - v_{rel})^\alpha) \quad (3.10)$$

kde  $\sigma$  je opět uživatelem zadaný parametr ovlivňující míru zkracování pružin mezi částicemi povrchu a pružin mezi částicemi žil a  $l^n$  je aktualizovaná délka pružiny mezi částicemi  $p_a$  a  $p_b$  a  $\alpha = 1$ .

Následně aktualizujeme polohy a rychlosti částic. Rychlosti částic se mohly měnit, pokud se nejedná o první snímek smyčky našeho algoritmu na deformace nebo je zahrnuto působení vnějších sil, jako je například gravitace:

$$\begin{aligned} v_i &= v_i + g\Delta t \\ x_i &= x_i + v_i\Delta t \end{aligned} \quad (3.11)$$

Ještě před tím, než začneme měnit polohy částic, musíme si dočasně uložit výšku voroinových částic vůči jejich příslušnému trojúhelníku:

$$h_i = (x_i^V - x_a^F)n \quad (3.12)$$

Jakmile jsou vypočítány nové délky pružin, aktualizovány polohy částic, jejich rychlosti a výšky voroinových částic dočasně uloženy, jsou pozice a rychlosti částic iteračně měněny pomocí algoritmu PBD [9] podle následujícího vztahu:

$$\begin{aligned} x_a &= x_a - \frac{w_1}{w_1 + w_2} C \frac{p_a - p_b}{\sqrt{(x_a - x_b)^2}} \\ x_b &= x_b + \frac{w_2}{w_1 + w_2} C \frac{p_a - p_b}{\sqrt{(x_a - x_b)^2}} \end{aligned} \quad (3.13)$$

kde  $x_a$  je poloha částice  $p_a$  a  $x_b$  je poloha částice  $p_b$ , které spojuje pružina  $s_j$ ,  $w_1 = 1/x_a$ ,  $w_2 = 1/x_b$  a  $C = \sqrt{(x_a - x_b)^2} - l_i$ .

Zároveň je aktualizována rychlost částic podle vztahu:

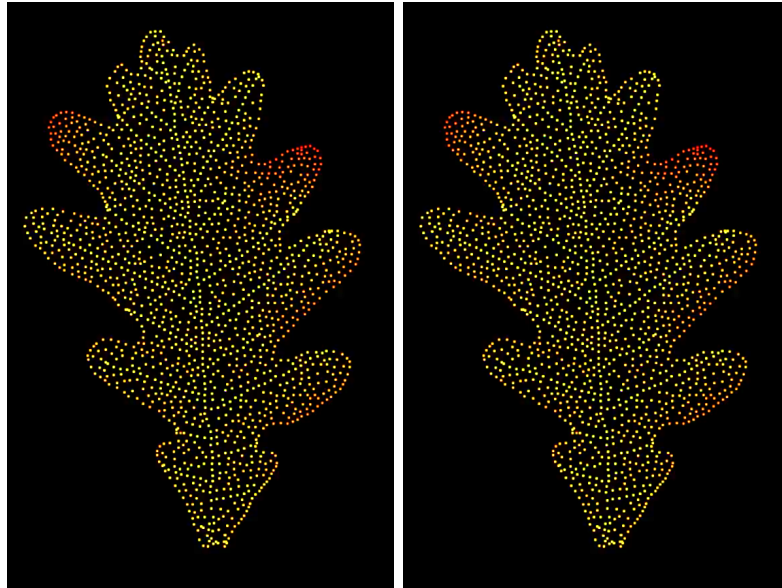
$$\begin{aligned} v_a &= v_a - \frac{w_1}{w_1 + w_2} C \frac{p_a - p_b}{t\sqrt{(x_a - x_b)^2}} \\ v_b &= v_b + \frac{w_2}{w_1 + w_2} C \frac{p_a - p_b}{t\sqrt{(x_a - x_b)^2}}. \end{aligned} \quad (3.14)$$

Na závěr jsou polohy všech voroinových částic  $p_i^V$  umístěny do středu jejich odpovídajícího trojúhelníku  $F$  ve výšce  $h_i$  vypočtené před algoritmem PBD a aktualizovány podle vzorce:

$$x_i^v = \frac{1}{3}(x_a^F + x_b^F + x_c^F) + h_i n \quad (3.15)$$

Jelikož splnění všech podmínek pro algoritmus PBD zmiňované Jeongem [6]  $|C/l| < \lambda_c$  nevyhází úplně podle našich představ, je počet PBD iterací navolen uživatelem. Tato skutečnost zapříčiňuje fakt, že jelikož jsou voroinovy částice každým snímkem aktualizovány do středu trojúhelníku  $F$ , nedojde nikdy ke kolizi. Z tohoto důvodu jsou detekce kolizí v této práci vynechány.

Celková hmotnost každé částice je tudíž aktualizována v každém snímku  $n$ . To, jak se hmotnost částice bude měnit, závisí na námi zvoleném parametru  $w_{loss}$ . Kdybychom nastavili parametr  $w_{loss} = 0$ , částice by nevysychaly, koncentrace vody vůči cytosolu by byla konstantní, délky pružin by se neměnily a tudíž by se neměnily ani pozice a hmotnosti jednotlivých částic. Jak je vidět na obr.3.13, žádná částice neztrácí svou hmotnost. Celá struktura se nachází v počátečním rovnovážném stavu a k žádnému pohybu částic nedochází.

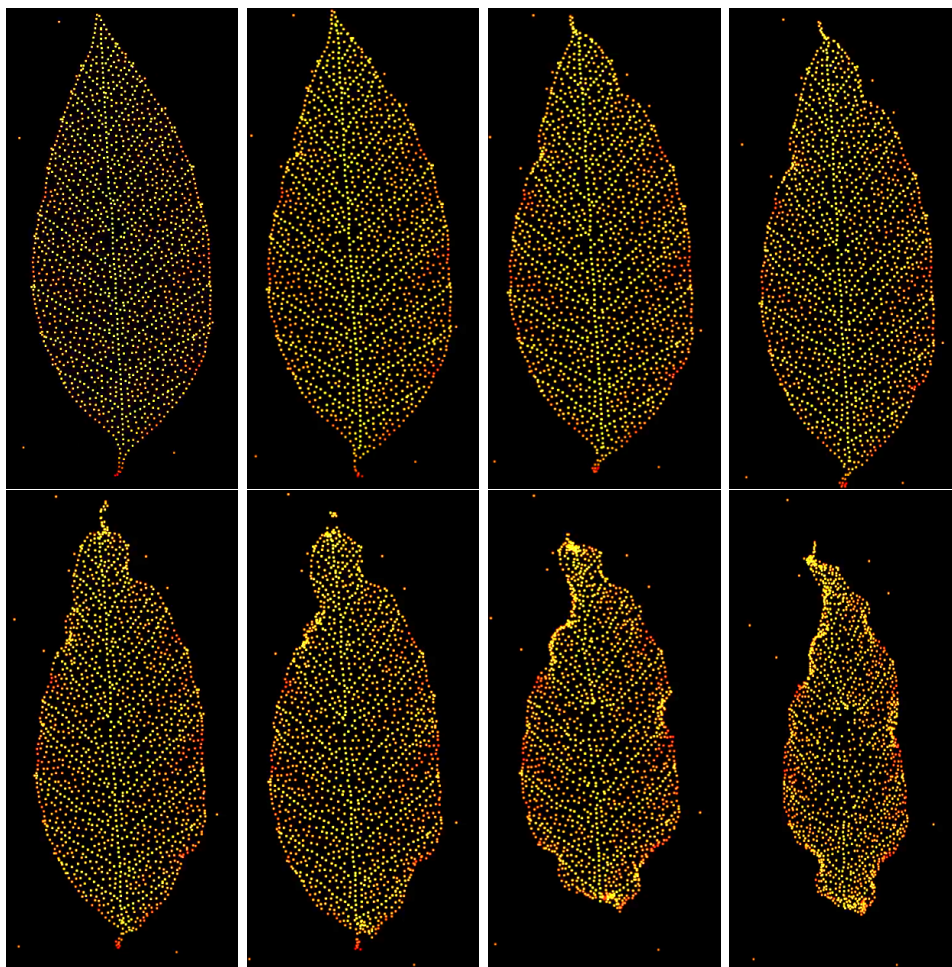


**Obrázek 3.13** : Hmotnost jednotlivých částic delaynovy vrstvy při nastaveném parametru  $w_{loss} = 0$ : obrázek vlevo vizualizuje hmotnost částic listu po inicializaci, obrázek vpravo v desáté snímku. Jak je vidět, změna není žádná.

Zadáme-li naopak parametr  $w_{loss}$  různý od nuly, začne částice na své

hmotnosti ubývat či přibývat v závislosti na znaménku parametru  $w_{loss}$ , rozpohybuje se (viz obr. 3.14) a díky tomu dochází k deformacím.

Větší hodnota tohoto parametru zapříčiňuje rychlejší vysychání každé částice, a tedy i rychlejší pohyb a výraznější deformace. Jakmile částice začínají vysychat, difúzní koeficient  $L$  částic začíná vycházet různý od nuly a mezi částicemi začíná proudit voda, aby se celá struktura listu dostala opět do rovnovážného stavu. Rychlost předávání vody mezi částicemi můžeme v naší implementaci ovlivnit ještě dodatečným parametrem  $\lambda$ , který zapříčiní, že se částice rychleji dostanou do rovnovážného stavu.



**Obrázek 3.14** : Pohyb jednotlivých částic delaynovy vrstvy a jejich hmotnost při nastaveném parametru  $w_{loss} > 0$ ,  $\sigma_{vein} = 0.5$   $\lambda = 70$  a  $\sigma_{surface} = 1.5$ , přidavném parametru ovlivňujícím rychlost difúze a 10 PBD iteracích pro každý snímek: obrázek vlevo vizualizuje hmotnost částic listu po inicializaci, každý obrázek napravo od něj pak vizualizuje polohy a hmotnost částic po zhruba sedmi snímcích.



## Kapitola 4

### Konkrétní implementace

V této kapitole se zaměříme na konkrétní implementaci žilek, které jsme potřebovali pro vygenerování listů a jejich procedurální deformace. Algoritmy se budeme snažit sumarizovat co nejstručněji, aby se vážený čtenář neutopil v pseudokódu, ale zároveň v takovém detailu, aby byly vystiženy zásadní myšlenky. Níže už uvádíme algoritmy pro generování žilek:

---

**Algorithm 1** Algorithm for generating main veins

---

```
1: for each main vein do
2:   while the target segment is not reached do
3:      $v_{gr} \leftarrow$  grow direction to the target final segment
4:      $s_{[i+1]} \leftarrow newVeinSegment(v_{gr}, \kappa, \omega, scale)$ 
5:      $growMainVein(s_{[i+1]})$ 
6:     if not the vein exceeds the threshold zone then
7:       mark segment as candidate segment for lateral veins
8:     end if
9:   end while
10: end for
```

---

---

**Algorithm 2** Algorithm for generating laterar and sublateral veins

---

```
1: for each vein candidate segment do
2:    $x \leftarrow$  number of segments to skip until we generate a lateral vein
3:   for each x-th segment do
4:     if line between start and target doesn't intersect anything then
5:       while the target segment is not reached do
6:          $s_{[i+1]} \leftarrow newVeinSegment(\kappa, scale)$ 
7:         if  $growVein(s_{[i+1]})$  intersects another vein or contour then
8:            $DeleteSegment(s_{[i]})$ 
9:           break
10:        end if
11:       end while
12:     end if
13:   end for
14: end for
```

---

Kde  $\kappa$  a  $\omega$  jsou parametry načtené z konfiguračního souboru udávající rozpětí náhodného vychýlení vektoru  $\vec{v}_i$  vedoucího z aktuálního segmentu žíly  $s[i]$  a vektor  $\vec{v}_{gr}$  působí ve směru cílového segmentu. Vygenerovaný segment  $s_{[i+1]}$  je vytvořen ve směru určeném průměrem vektorů  $\vec{v}_{gr}$  a  $\vec{v}_i$  a náhodně zvětšen pomocí parametru  $scale$ , aby byl každý úsek žíly jinak dlouhý.

Jak můžeme vidět, používáme dva různé algoritmy. A to algoritmus pro generování hlavní žíly a algoritmus pro generování laterálních a sublaterálních žilek. Generování laterálních a sublaterálních žilek se od generování hlavní žíly liší pouze tím, že nebere v úvahu globální úhel, ve kterém se hlavní žíla během růstu vychyluje a testuje se na kolize jak s ostatními již vygenerovanými žilami, tak s konturou listu.

Posledním algoritmem je algoritmus pro osmotický tok a deformace, který následuje kroky v námi navrženém řešení a vypadá následovně:

---

**Algorithm 3** Algorithm for generating deformations
 

---

```

1: for each frame n do
2:   for each particle  $p_i^{T \cup V}$  do
3:      $L_i \leftarrow \text{calculateDiffusion}(s_{i_{neighbours}}, p_{i_{neighbours}})$ 
4:   end for
5:    $w^T \leftarrow \text{osmoticWaterFlow}(L, p^T, s^T, w_{loss}^T)$ 
6:    $w^V \leftarrow \text{osmoticWaterFlow}(L, p^V, s^V, w_{loss}^V)$ 
7:   for each spring  $s_j^{T \cup V \cup L}$  do
8:      $l_j \leftarrow \text{shrinkSpringLength}()$ 
9:   end for
10:  for each particle  $p_i^{T \cup V}$  do
11:     $v_i \leftarrow v_i + g\Delta t$ 
12:     $x_i \leftarrow x_i + v_i\Delta t$ 
13:  end for
14:  for each particle  $p_i^V$  do
15:     $h_i \leftarrow \text{calculate length of } p_i^V \text{ projection to the triangle } F$ 
16:  end for
17:  for each projection step do
18:     $x^T, v^T \leftarrow \text{PBD}(x^T)$ 
19:     $x^V, v^V \leftarrow \text{PBD}(x^V)$ 
20:     $x^{T \cup V}, v^{T \cup V} \leftarrow \text{PBD}(x^{T \cup V})$ 
21:  end for
22:  for each particle  $p_i^V$  do
23:     $x_i^V \leftarrow (x_a^F + x_b^F + x_c^F)/3 + h_i n_i$ 
24:     $v_i^V \leftarrow (v_a^F + v_b^F + v_c^F)/3$ 
25:  end for
26: end for

```

---

kde  $s_{i_{neighbours}}$  udává všechny pružiny, které vedou z dané částice  $p_i$  a  $p_{i_{neighbours}}$  udává všechny sousední částice, do kterých vede nějaká z pružin  $s_{i_{neighbours}}$ .



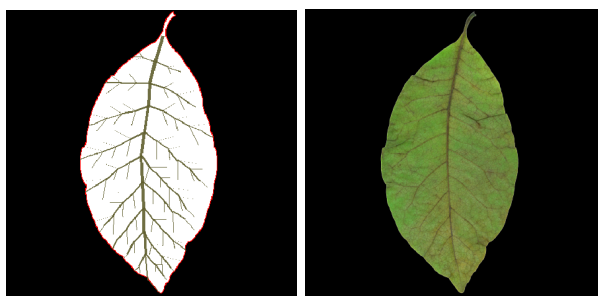
## Kapitola 5

### Výsledky

V této kapitole odprezentujeme dosažené výsledky. Nejdříve ukážeme vygenerované žilky listů, které porovnáme s reálnými listy a listy z popisovaných metod a následně ukážeme výsledky z deformací, které opětovně porovnáme s reálnými listy a výsledky existujících metod. Obrázky 5.1 a 5.2 znázorní porovnání naší vygenerovaných žil pro list javoru a ořechu. V těchto případech jsme zapli i generování sublaterálních žilek. V některých případech však mají listy jednodušší strukturu žilek, a tak sublaterální žilky vykreslovat nebudeme.

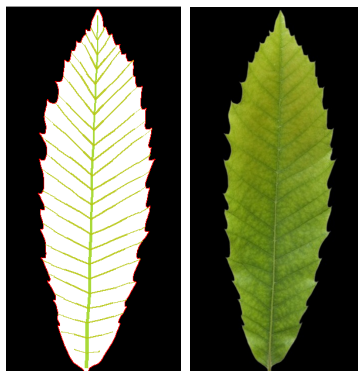


**Obrázek 5.1 :** List javoru: vlevo vidíme vygenerované žilky dosažené při nastavení konfiguračních parametrů lokálního úhlu  $\kappa = 5$ , globálního úhlu  $\omega = 2$  a vhodného nastavení parametrů profilu, vpravo pak reálný list javoru.



**Obrázek 5.2 :** List ořešáku: vlevo vidíme ukázkou žilek vygenerovaných naším programem při vypnutí tvorby sublaterálních žilek a nastavení konfiguračních parametrů úhlu  $\kappa = 7$ ,  $\omega = 10$  a vhodného nastavení parametrů profilu, vpravo list ořešáku.

Můžeme si všimnout, že pro pravidelné typy listů (viz obr. 5.3 a obr. 5.4), pro které nevykresluje sublaterální žilky, se přibližujeme reálným výsledkům. To je umožněno díky konfiguračnímu souboru, ve kterém můžeme kromě generování profilu listů a tloušťky žil ovlivňovat také hodnotu úhlu, který můžeme snížit na minimální hladinu, případně jej zcela vyřadit.



**Obrázek 5.3 :** List kaštanovníku: vlevo vidíme žilky vygenerované naším programem při vypnutí tvorby sublaterálních žilek, nastavení konfiguračních parametrů úhlu  $\kappa = 2$ ,  $\omega = 1$  a vhodného nastavení parametrů profilu, vpravo pak vidíme ukázkou kaštanovníku jedlého, který má pouze hlavní a laterální žilky.



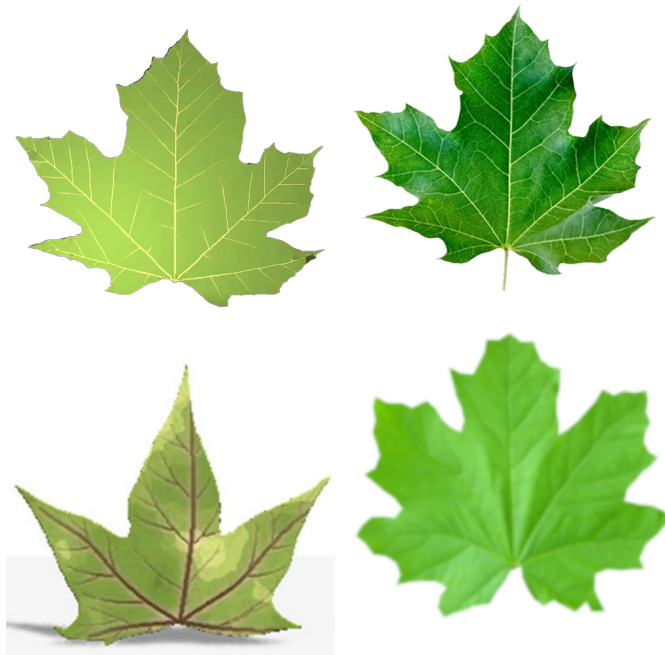
**Obrázek 5.4 :** List dubu: vlevo vidíme žilky vygenerované naším programem při nastavení konfiguračních parametrů úhlu  $\kappa = 4$ ,  $\omega = 1$  a vhodného nastavení parametrů profilu, vpravo pak reálný dubový list.

Poté, co máme vygenerované žilky pro daný list, můžeme žilky zkombinovat spolu s barvou listu díky míchání barev v OpenGL. Vzniklé vygenerované listy pak porovnáváme s referenčním obrázkem a zmiňovanými metodami pro generování listů (viz. obr. 5.5, obr. 5.6, obr. 5.7 a obr. 5.8). Poslední částí výsledků, které porovnáváme s referenčním obrázkem a zmiňovanými metodami, jsou deformace. Na obrázku 5.9 a 5.10 si můžeme připomenout výsledky dosažené předchozími studii. Námi dosažené deformace jsou vizualizovány na obrázcích 5.11, 5.12, 5.13 a 5.14

Závěrečná tabulka vizualizuje zvolené parametry pro vymodelované listy (viz tab. 5.1), a udává i množství jednotlivých částic a pružin daného listu.



**Obrázek 5.5 :** Porovnání vygenerovaných listů líčidla amerického na levé straně s reálným listem uprostřed a podobného listu (převzato z: [1]) vygenerovaného metodou pro generování listů pomocí žilek od Kima.



**Obrázek 5.6 :** Porovnání vygenerovaných listů dubu s reálným listem: obrázek vlevo nahoře zobrazuje námi vygenerovaný list javoru, obrázek vpravo nahoře vizualizuje reálný list javoru, obrázek vlevo dole ukazuje vygenerovaný list javoru metodou pro generování listů pomocí žilek (převzato z: [1]) a list vpravo dole je výsledkem metody generování listů pomocí implicitní funkce (převzato z: [4]).



**Obrázek 5.7 :** Porovnání vygenerovaných listů dubu s reálným listem: vlevo vidíme námi vygenerovaný list dubu, uprostřed reálný list dubu a napravo list dubu vygenerovaný pomocí metody generování listů pomocí implicitní funkce od Pinedy (převzato z:[4]).



**Obrázek 5.8 :** Porovnání vygenerovaného listu kaštanovníku jedlého s reálným listem: vlevo vidíme námi vygenerovaný list kaštanovníku, vpravo pak reálný list.



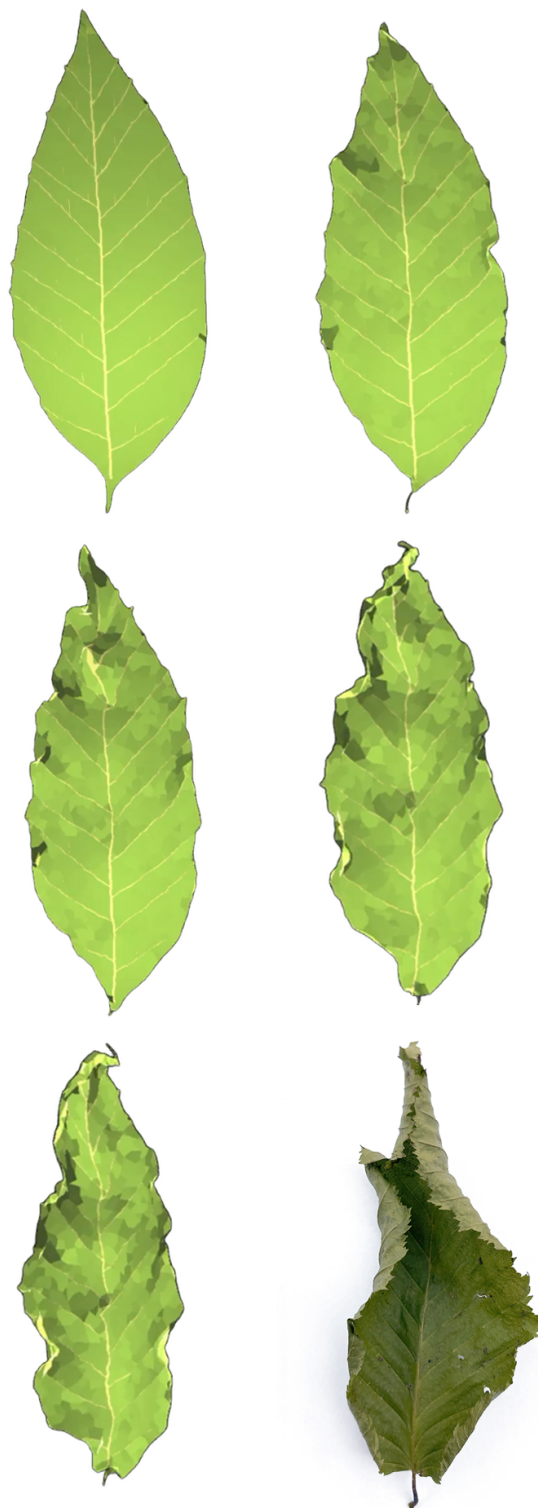
**Obrázek 5.9 :** Deformace listů vygenerované metodou založené na hlavních žilkách (převzato z: [5]).



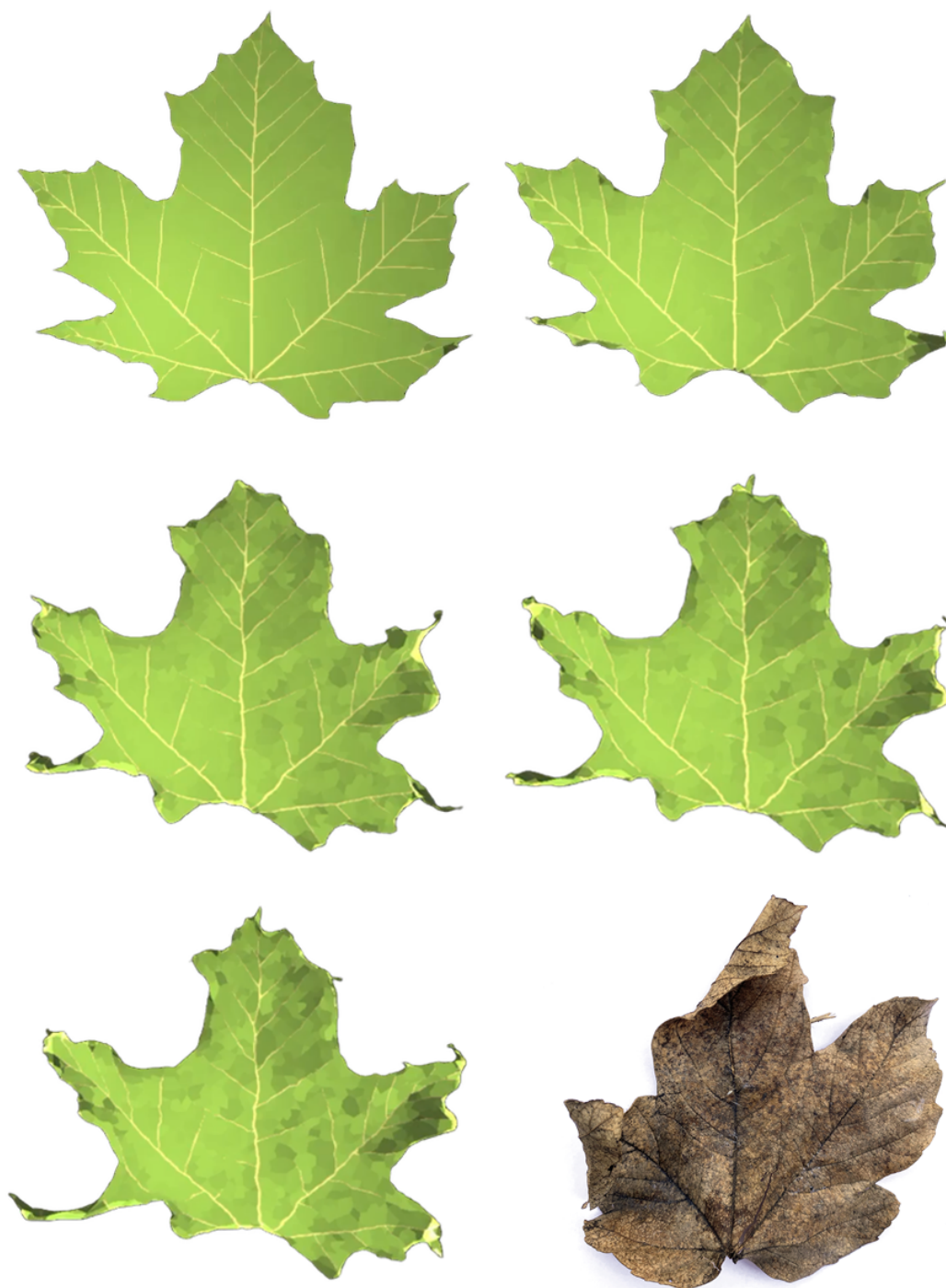
**Obrázek 5.10** : Deformace listů vygenerované metodou založené na žilkách (převzato z: [6]).

	Javor	Kaštanovník	Dub	Líčidlo	Všechny dohromady
$\Delta t$					0.002
$g$					-9.8
$D$					1
$\alpha$					1
$\sigma_{vein}$					0.5
$\sigma_{surface}$					1.5
$l_{min}$					0.1
$m_{min}$					0.1
$m_{max}$					1.0
PBDiter.	1	1	10	1	
$W_l^T_{oss}$	-0.1	-0.08	-0.1	-0.1	
$W_l^V_{oss}$	-0.2	-0.15	-0.2	-0.2	
$\lambda$	70	700	700	700	
Počet částic	5674	3355	3721	3634	
Počet pružin	21457	12471	13759	13434	

**Tabulka 5.1** : Parametry vymodelovaných listů a počet částic a pružin.



**Obrázek 5.11** : Ukázka námi vygenerovaných deformací líčidla amerického: Vpravo dole je vizualizován reálný list podobného typu, nalevo a nahoře od něj jsou vizualizovány postupné deformace tohoto listu.



**Obrázek 5.12 :** Ukázka námi vygenerovaných deformací listu javoru: Vpravo dole je vizualizován reálný list javoru, nalevo a nahoře od něj jsou vizualizovány postupné deformace tohoto listu.



**Obrázek 5.13 :** Ukázka námi vygenerovaných deformací listu dubu: Vpravo dole je vizualizován reálný list dubu, nalevo a nahoře od něj jsou vizualizovány postupné deformace tohoto listu.





**Obrázek 5.14 :** Ukázka námi vygenerovaných deformací listu kaštakovníku: Vpravo dole je vizualizován reálný list podobného tvaru, nalevo a nahoře od něj jsou vizualizovány postupné deformace tohoto listu.





## Kapitola 6

### Diskuse

Podářilo se nám propojit metody generování žilek spolu s metodou deformací a její výstupy vizualizovat, což bylo primárním cílem této práce. Výsledky se však potýkají s nedokonalostí a nepřesností, a to jak při generování žilek, tak při samotných deformacích. Nepřesnost žilek je zapřičiněna ne zcela přesnými nakonfigurovanými hodnotami v konfiguračním souboru a nepřesné deformace nedotažením algoritmu PBD a pravděpodobně absencí detekcí kolizí. Lepších výsledků žilek bychom mohli dosáhnout drobnou úpravou systému jejich generování (například lepší způsob pro určení úhlu růstu) či automatickou detekcí kandidátních bodů na cílové segmenty z kontury. Více realistických deformací bychom dosáhli při dotažení algoritmu PBD do samotného konce - respektive jeho omezujících podmínek a doimplementováním s ním souvisejících detekcí kolizí.





## Kapitola 7

### Závěr

V této práci jsme čtenáře obeznámili s existujícími studii generování listí, procedurálních deformací a u nejvíce zajímavých či použitých metod jsme podrobně popsali implementaci z těchto studií vycházejících a námi doplněných. V této práci jsme se inspirovali metodou pro generování listí pomocí žilek a navázali na ni metodou procedurálních deformací.

Program byl vyvíjen za použití operačního systému Windows 10 ve vývojovém prostředí Visual Studio 2019 a Visual Studio 2022. Vizualizace výsledků žilek a hmoty částic probíhala za použití knihovny CImg, vizualizace listí a jejich deformací za použití OpenGL. Program byl také vyzkoušen jak na operačním systému macOS po zprovoznění knihovny cImg, tak i na tomto systému. Počítač použitý na vývoj tohoto programu měl procesor Intel(R) Core(TM) i7-5930K CPU @ 3.50GHz, 28 GB RAM a grafickou kartu AMD Radeon Graphics Processor (0x67DF). Počítač použitý na testování měl procesor Intel Core 2 Duo 3,06 GHz, 4GB RAM a grafickou kartu ATI Radeon HD 4670 256 MB.





## Literatura

- [1] Daeyeoul Kim and Jinmo Kim *Procedural modeling and visualization of multiple leaves*. Publisher X, 22 February 2016.
- [2] Martin Pažout *Semestral Project*, January 2021.
- [3] Monssef Alsweis, Oliver Deussenn and Jia Liu *Simulation and visualization of adapting venation patterns*. Department of Computer Information Science, University of Konstanz, Konstanz, Germany, 2 SIAT, Chinese Academy of Science, Shenzhen, China, 3 School of Automation, Beijing Information Science Technology University, Beijing, China, January 2021.
- [4] Israel Pineda and Oubong Gwun *Leaf Modeling and Growth Process Simulation Using the Level Set Method*. Department of Computer Science and Engineering, Chonbuk National University, Jeonju 54896, South Korea, August 11, 2017.
- [5] Xiaopeng Sun, Jia Fu ,Teng Chen, Yu Dong *Wrinkle and curl distortion of leaves using plant dynamic*. School of Electronic Information Engineering, Foshan University, China, School of Computer and Information Technology, Liaoning Normal University, China 2021
- [6] SoHyeon Jeong, Si-Hyung Park and Chang-Hun Kim *Simulation of Morphology Changes in Drying Leaves*. Department of Computer and Radio Communications Engineering, Korea University, Korea, Volume 32 (2013), number 1 pp. 204–215
- [7] *David H. Douglas and Thomas K. Peucker: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature*. *Cartographica*. 10, 112–122 (1973).
- [8] *Jinmo Kim : Modeling of various digital leaves using feature-based image warping*. *J. Digit. Contents Soc* 16(2), 235–244 (2015).
- [9] *Matthias Müller, Bruno Heidelberger, Marcus Hennix and John Ratcliff: Position based dynamics*. *Journal of Visual Communication and Image Representation* 18, 2 (April 2007), 109–118.
- [10] *Jana Křečková: Morfologické znaky jednoduchých listů dřevin - Bakalářská práce, Praha 2014*.





## Příloha A

### Obsah přiloženého média

readme.txt .....	instalační příručka
├─ src .....	zdrojový adresář
│   └─ images .....	adresář s obrázky
│   └─ JSON	
│       └─ RSJparser .....	parser konfiguračního souboru
│   └─ main .....	hlavní program
│   └─ latex .....	adresář pro zdrojové kódy latexu
└─ generated	
└─ thesis.pdf .....	PDF verze bakalářské práce