Na tomto místě bude oficiální zadání vaší práce

- Toto zadání je podepsané děkanem a vedoucím katedry,
- musíte si ho vyzvednout na studiijním oddělení Katedry počítačů na Karlově náměstí,
- v jedné odevzdané práci bude originál tohoto zadání (originál zůstává po obhajobě na katedře),
- ve druhé bude na stejném místě neověřená kopie tohoto dokumentu (tato se vám vrátí po obhajobě).

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačové grafiky a interakce

Diplomová práce

Ilustrace pohybu mechanických součástí

Bc. Štětovská Markéta

Vedoucí práce: Ing. Ladislav Čmolík, Ph.D.

Studijní program: Otevřená informatika, strukturovaný, Navazující magisterský

Obor: Počítačová grafika

30. prosince 2013

Poděkování

Chtěla bych poděkovat panu doktoru Čmolíkovi za jeho cenné rady a trpělivý a vstřícný přístup. Dále bych chtěla poděkovat svým rodičům za podporu během mého studia.

Prohlášení

Prohlašuji, že jsem práci vypracovala samostatně a použila jsem pouze podklady uvedené v přiloženém seznamu.
Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 30. 12. 2013 .....

Abstract

The task of this work is to analyze possible ways to illustrate the function of the system, which consists of several mechanical parts, e.g., a system of gears, and to design and implement application that will create a static illustration of the movement of mechanical parts by adding graphical annotations in the form of arrows, and animate the movement of the system of mechanical parts based on user interaction with one of the mechanical parts.

The contribution of this thesis is to create software that automatically generates dynamic and static visualization of three-dimensional objects representing mechanical parts of a system, that minimizes required user interaction.

Abstrakt

Úkolem práce je analyzovat možné způsoby ilustrace funkce soustav skládajících se z více mechanických součástí (např. soustava ozubených kol) a na základě analýzy navrhnout a implementovat aplikaci, která umožní vytvořit statickou ilustraci pohybu mechanických součástí pomocí přidání grafických anotací ve formě šipek a animovat pohyb soustavy mechanických součástí na základě pohybu uživatele s jednou vybranou částí.

Přínosem práce je vytvoření aplikace, která automaticky vygeneruje na základě načtených třídimenzionálních objektů, které představují mechanické součásti soustavy, dynamickou a statickou vizualizaci, a která minimalizuje potřebné množství interakce uživatele.

Obsah

1 Úvod **1**
1.1 Statická ilustrace . . . 2
1.2 Cíle práce . . . 6
1.3 Struktura práce . . . 6

2 Analýza a návrh **7**
2.1 Určení vlastností součásti . . . 9
2.1.1 Nalezení osy symetrie objektu . . . 11
2.1.2 Nalezení rotační osy 3D objektu . . . 11
2.1.3 Realizovaná metoda . . . 12
2.2 Nalezení součástí soustavy, které spolu mají kontakt . . . 19
2.2.1 SAT . . . 20
2.2.2 BVH . . . 20
2.2.3 Minkowského součet . . . 24
2.2.4 Detekce kolizí založena na voxelizaci . . . 24
2.2.5 Voxelizace . . . 28
2.3 Vytvoření grafu interakce . . . 34
2.4 Dynamická vizualizace soustavy mechanických součástí . . . 39
2.4.1 Výpočet rychlosti rotace součástí soustavy, která neobsahuje nepohyblivou součást . . . 41
2.4.2 Stanovení poloměru roztečné kružnice součásti . . . 44
2.4.3 Stanovení směru rotace součásti . . . 47
2.4.4 Definice typů součástí . . . 47
2.4.5 Výpočet rychlosti rotace ramene okolo rotační osy nepohyblivé součásti 48
2.4.6 Rozklad grafu interakce na podgrafy a stanovení, zda je animace soustavy proveditelná . . . 51
2.4.7 Aliasing dynamické vizualizace soustavy mechanických součástí . . . 55
2.4.8 Omezení dynamické vizualizace soustavy mechanických součástí . . . 56
2.5 Statická vizualizace . . . 56
2.5.1 Kauzální řetězec . . . 57
2.5.2 Klíčové snímky . . . 59
2.5.3 Šipky . . . 59
2.6 Shrnutí . . . 74

3 Realizace **77**
3.1 Nalezení vlastností součásti 78
3.2 Nalezení součástí soustavy, které spolu mají kontakt 80
3.3 Vytvoření grafu interakce 81
3.4 Vytvoření dynamické vizualizace soustavy mechanických součástí 81
3.5 Statická vizualizace 82

4 Testování **83**
4.1 Výsledky testování vlastností součásti 83
4.2 Výsledky testování grafu interakce součásti 85
4.3 Výsledky testování statické ilustrace se šipkami 87

5 Závěr **97**

A Instalační a uživatelská příručka **103**

B Konvexní obálky **105**
B.1 Jarvisův algoritmus 106
B.2 Quick hull 108
B.3 Rozděl a panuj 108
B.4 Quick hull ve 3D 108

C Výsledky testování **111**

D Obsah přiloženého CD **125**

Kapitola 1

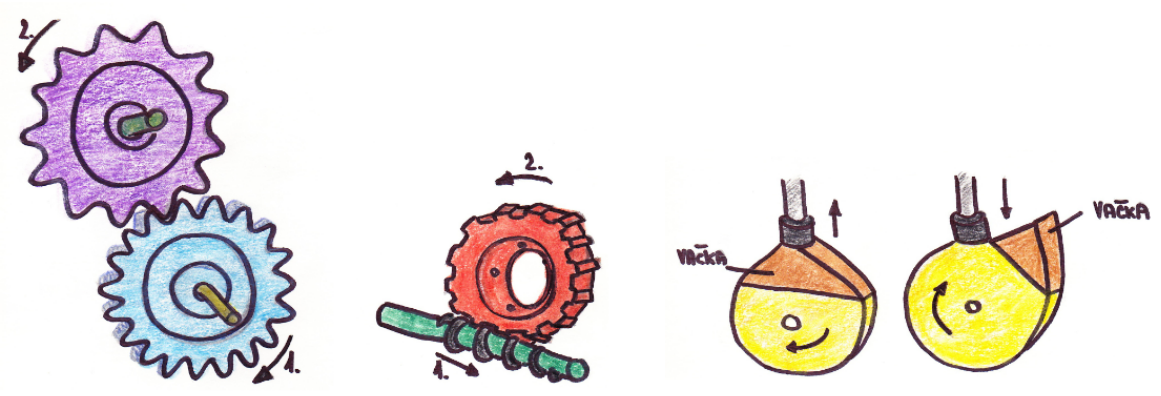
Úvod

Vizualizace pomáhá lidem lépe pochopit význam zobrazovaných dat a vztahy mezi nimi, protože umožňuje reprezentovat zobrazovaná data ve formě, ve které se mohou uplatnit schopnosti vizuálního systému člověka. Při vytváření vizualizace je proto potřeba zohlednit vlastnosti vizuálního systému člověka. Vizualizace poskytuje místo existující numerické reprezentace vizuální reprezentaci dat. Vizualizační techniky jsou založené na schopnosti člověka vnímat (z angl. perception) a rozpoznávat (z angl. cognitive) vizuální informace. Zpracování vizuální informace je v prvních přibližně 200 ms provedeno bez přičinění našeho vědomí. Tato hodnota odpovídá rychlosti pohybu očí. Nevědomé zpracování dat [22] představuje sběr informací získaných z vizuální reprezentace dat bez přičinění našeho vědomí.

Existují dva základní druhy vizualizace, jimiž je vědecká vizualizace a vizualizace informace. Vědecká vizualizace se zabývá znázorněním třídimenzionálních jevů (např. architektonických, lékařských, meteorologických) se zaměřením se na jejich realistické vyjádření (např. objemu, povrchu). Vizualizace informace představuje vizuální reprezentaci abstraktních dat za účelem posílení kognitivního vnímání člověka. Abstraktní data obsahují jak numerickou tak grafickou (např. textovou) informaci. Zjednodušeně lze říci, že umístění dat v prostoru pro vědeckou vizualizaci je dáno definicí dat a že pro vizualizaci informace dochází ke zvolení vhodného umístění znázorňovaných informací. Tato práce se zabývá vědeckou vizualizací pohybu soustavy mechanických součástí v třídimenzionálním prostoru.

Data, jejichž vizuální reprezentace má být zobrazena, prochází vizualizačním řetězcem, který se skládá obecně ze tří základních kroků. Prvním krokem řetězce je výběr požadovaných dat, které mají být zobrazeny. Druhý krok řetězce představuje mapování dat na jejich geometrickou reprezentaci (např. šipky, barvy), které je možné zobrazit. Posledním krokem řetězce je vykreslení stanovené geometrické reprezentace.

Cílem této práce je implementovat podmnožinu technik, které uvádějí Mitra a kol. v článku *Illustrating How Mechanical Assemblies Work* [27]. Článek popisuje, jak vytvořit vizualizaci pohybu soustavy mechanických součástí, která slouží pro pochopení funkce zobrazené soustavy. Soustava mechanických součástí je složena z jedné či několika součástí, které mají v třídimenzionálním prostoru určité prostorové uspořádání. Jedna ze součástí soustavy, která uvede soustavu do pohybu, se nazývá řídící součást. Řídící součást může být připojena například ke krokovému motoru či k jinému zařízení, jehož rychlost a směr otáčení je možné libovolně měnit. Znázornit pohyb soustavy mechanických součástí lze buď pomocí statické



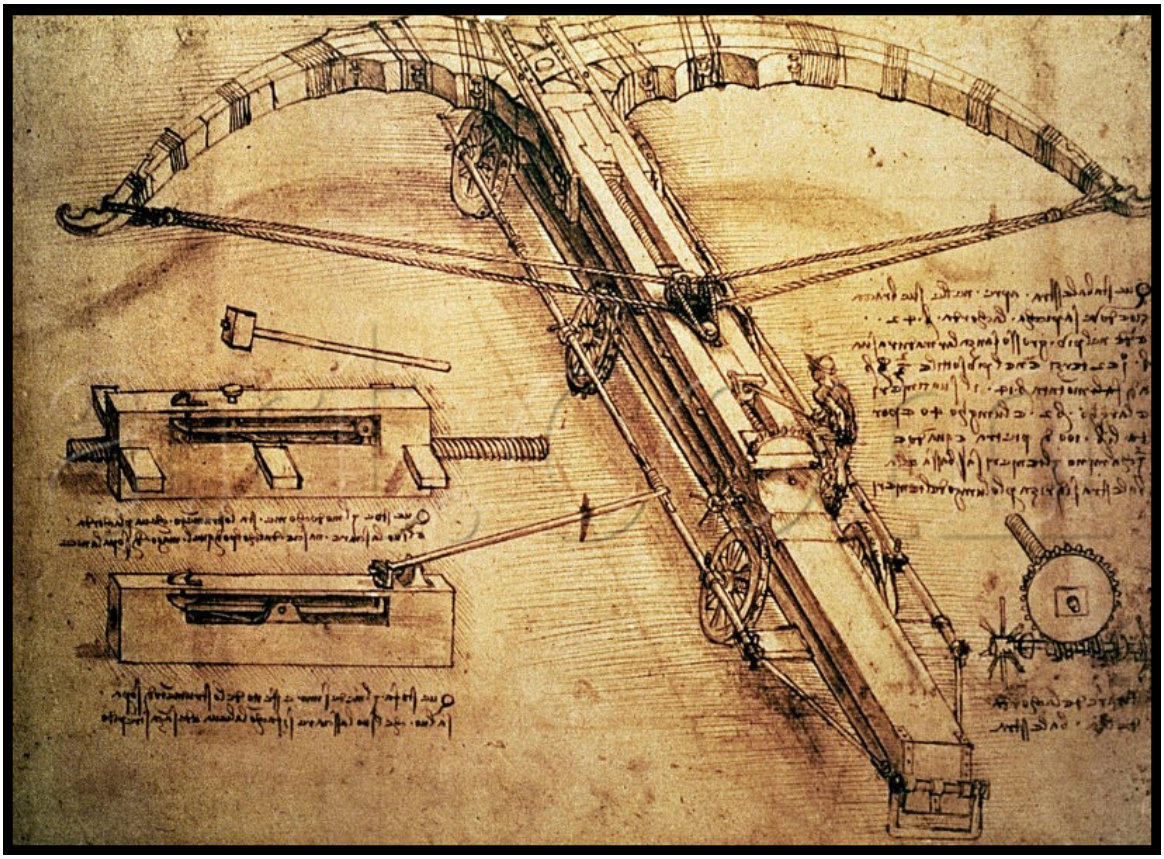
a) ozubené soukolí b) šnekové soukolí c) znázornění pohybu vačky

Obrázek 1.1: Příklady použití techniky pomocných informací pro ručně kreslenou statickou ilustraci pohybu soustavy mechanických součástí: (a,b) šipky, (c) sekvence časových snímků.

ilustrace s využitím pomocných informací (např. Mitra a kol. [27] prezentují techniku vytvoření šipek a posloupnosti časových snímků – obr. 1.1) a nebo pomocí dynamické vizualizace. Dynamickou vizualizaci je vhodné použít v případě, kdy by statická ilustrace neposkytla dostatečnou informaci o složitějším pohybu soustavy mechanických součástí. Podmnožina technik realizovaná v této práci zahrnuje určení vlastností objektu, vytvoření dynamické vizualizace a statické ilustrace pohybu soustavy mechanických součástí s využitím pomocné grafické informace ve formě šipek, které znázorňují směr pohybu součástí soustavy.

1.1 Statická ilustrace a dynamická animace pohybu

Stopy používání techniky statické ilustrace s pomocnými vizualizačními technikami sahají až do starší doby kamenné, kdy tamější lovci zaznamenávali techniku lovu mamuta pomocí jeskynních nástěnných maleb. Techniku lovu mamuta znázorňovali buď vrytím obrysů například pomocí pazourku nebo malbou na stěnu použitím dřevěného uhlí, zeminy či krve ulovené zvěře. Již v tehdejší době lidé věděli, že bylo potřeba pečlivě naplánovat důležité kroky, které byly zásadní pro úspěšnost lovu. K výraznému rozvoji ilustrace došlo v období renesance. Metoda statické ilustrace s pomocným textem, který vysvětloval funkci zobrazeného objektu, se používala například pro nakreslení náčrtku konstrukce a vysvětlení funkce nových vynálezů. Na obrázku 1.2 je znázorněn náčrtek obrovského samostřílu, který vynalezl Leonardo da Vinci. Leonardo da Vinci žil v období renesance a byl mimo jiné konstruktérem, vynálezcem a malířem. Tento obrovský samostříl sloužil k vystrašení protivníka svojí velikostí. Zařízení mělo tři páry kol, které umožňovaly jeho přesun. Toto zařízení bylo spíše konstruováno pro střílení velkých kamenů nebo hořících předmětů. Když měl samostříl vystřelit, tak se udeřilo palicí do kolíku, který tvořil spouštěcí mechanismus a který držel tětivu samostřílu napnutou. S vývojem lidstva dochází k vytváření složitějších předmětů, které se stávají běžnou součástí našeho života. Čím složitější jsou znázorňované předměty, tím pro-



Obrázek 1.2: Náčrt konstrukce samostřílu Leonardem da Vinci. Zdroj obrázku [13].

pracovanější techniky ilustrace je potřeba vytvořit a tím potřebnější je vytvoření ilustrace, aby bylo zřejmé, jak znázorněný předmět funguje. S nástupem počítačů došlo k vytvoření digitální (počítačové) ilustrace. Existují obecně dva základní nástroje pro vytvoření digitální ilustrace, jimiž jsou rastrové (bitmapové) a vektorové aplikace. V rastrové aplikaci je obrázek reprezentován pravoúhlou pravidelnou mřížkou, která je tvořena pixely. Digitální ilustrace vytvořená pomocí vektorového programu se skládá ze základních geometrických útvarů, jimiž jsou například přímky, mnohoúhelníky a body. Ke zhoršení kvality obrázku dochází z uvedené dvojice typů obrázků pouze pro jeho rastrovou reprezentaci z důvodu uložení informace do rastru. Vektorový obrázek má obecně menší paměťové nároky nežli rastrový obrázek. Vektorový obrázek není vhodné použít pro zápis fotografie, protože grafické objekty obsažené ve fotografii mají příliš složitou geometrii, a proto by měla vektorová grafika větší paměťové a výpočetní nároky nežli rastrová grafika.

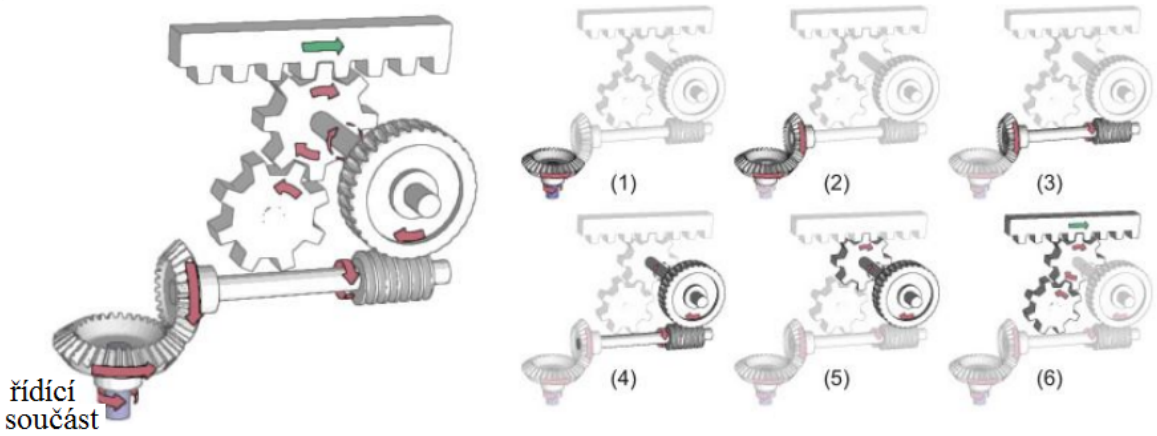
Obecně můžeme rozdělit vizualizaci pohybu soustavy mechanických součástí do dvou skupin. První skupinou je statická ilustrace a druhou skupinou je dynamická vizualizace pohybu. Statická ilustrace zobrazuje soustavu z takového definovaného pohledu, aby nejlépe zobrazoval textem popisovanou funkci soustavy. Dynamická vizualizace umožňuje manipulaci uživatele se soustavou (např. otáčení, změna měřítka, posun, zvolení oblasti zájmu) při

vizualizaci jejího pohybu.

Pokud se jedná o složitější soustavu mechanických součástí, která je zobrazena pomocí statické ilustrace, pak divák nemusí být schopen mentálně zrekonstruovat, jak změna polohy jedné mechanické součásti v dané soustavě změní polohu ostatních mechanických součástí uvnitř této soustavy. Existuje množství pomocných vizualizačních technik pro statickou ilustraci (např. Mitra a kol. [27] prezentují techniku šipek a sekvenci časových snímků – obrázek 1.1), které pomohou uživateli lépe mentálně zrekonstruovat, jak daná soustava mechanických součástí funguje. Pomocná vizualizační technika šipek dodá uživateli informaci o směru pohybu dané mechanické součásti. Pohyb většiny mechanických součástí je periodický. Rychlost pohybu mechanické součásti se však může během jedné periody pohybu měnit. Může také například docházet k tomu, že součást bude během vykonávání svého pohybu ztrácet a získávat kontakt s ostatními součástmi. Pro zobrazení takovýchto složitějších pohybů je vhodné použít pro statickou ilustraci techniku sekvence časových snímků. Mitra a kol. [27] prezentují techniku sekvence časových snímků (viz obrázek 1.1(c)), která vychází z poznatku, že je důležité zachytit klíčové části pohybu, které jsou zásadní pro pochopení, jak bude probíhat animace pohybu soustavy, a které by uživatel nedokázal získat z jednoho definovaného pohledu na zobrazovanou soustavu za účelem mentální rekonstrukce jejího pohybu. Pomocí statické ilustrace není pro uživatele jednoduché si představit přesnou časovou souslednost pohybu jednotlivých mechanických součástí uvnitř dané soustavy, jak rychle se dané mechanické součásti pohybují, popřípadě jak dochází k návaznosti pohybu jednotlivých mechanických součástí v závislosti na jejich předchozím uspořádání v prostoru.

Jednou z možností, jak lze ilustrovat pohyb soustavy mechanických součástí, je využití kauzálního řetězce (viz obrázek 1.3). Mitra a kol. [27] prezentují techniku vytvoření kauzálního řetězce, který představuje chronologicky uspořádanou posloupnost událostí, ve které vznik jedné události řetězce vyvolá jednu či více dalších událostí. Událostí se rozumí udělení pohybu součásti, která je součástí soustavy. Ze zobrazených kroků kauzálního řetězce vyplývá, že v každém kroku je v kauzálním řetězci zvýrazněno právě tolik součástí, kolik jich má kontakt se součástí, která jim předává svoji kinetickou energii, a je zvýrazněna také součást předávající energii.

V době, kdy vývoj techniky byl ještě v počátcích a výpočetní technika nebyla tak rozšířena, byly statické ilustrace prakticky jediným prostředkem, jak znázornit uživateli, jak daná vyobrazená soustava funguje. Pro vytvoření kvalitní statické ilustrace je důležité, aby byl zvolen takový vhodný pohled na zobrazenou soustavu, který bude co nejlépe znázorňovat její vysvětlovanou funkci. V současné době je výpočetní technika (např. stolní počítače, notebooky, tablety, smartphony) součástí běžného života téměř každého moderního člověka. Technické pokroky, které způsobily, že zařízení výpočetní techniky jsou přenositelnější, flexibilnější, výkonnější a uživatelsky přívětivější, se nezadržitelně vyvíjejí kupředu. S nadsázkou by se dalo říci, že v posledních dvou letech začíná výpočetní technika pomalu zastávat funkci papíru. Člověk místo toho, aby měl půjčenu či zakoupenu knihu, tak má elektronickou čtečku. Dokonce i v knihovnách se v průběhu posledního roku začínají půjčovat elektronické čtečky vedle vypůjčování samotných knih. Místo psaní poznámek na papír či do diáře se zaznamenávají poznámky do „chytrých“ telefonů, které nezastávají pouze základní funkci mobilního zařízení, ale umožňují například surfovat po internetu, číst soubory různých formátů, sledovat filmy, skládat hudbu, programovat. Za pár let budou tablety disponovat stejnou funkcionalitou, jakou dnes mají počítače. Dalo by se tedy říci, že se všudy přítomnou technikou dochází



Obrázek 1.3: (vlevo) Soustava mechanických součástí s využití pomocné grafické informace (šipek). (vpravo) Dílčí kroky kauzálního řetězce soustavy mechanických součástí. Texty obrázku byly přeloženy do češtiny autorkou práce. Zdroj obrázku [27].

k nahrazování statické ilustrace dynamickou vizualizací. Pomocí dynamické vizualizace uživatel získá lepší prostorovou představu o chování zobrazované soustavy.

Nyní si definujeme typy uživatelských rolí, které budou použity v následujícím textu. Uživatelská role nazvaná jako *expert* představuje takového člověka, který má odborné znalosti uvažované oblasti, avšak nemusí vědět, jak má tyto znalosti prezentovat ostatním lidem. *Ilustrátorem* je člověk, který má dovednost, že umí dobře kreslit, a ví, jak korektně vizuálně prezentovat nějaké znalosti ostatním lidem. Ilustrátor by měl mít také alespoň nějaké odborné znalosti z dané oblasti. *Grafikem* je takový člověk, který má odborné znalosti z oblasti počítačové grafiky a animace. Uživatelskou rolí *programátora* označíme takového člověka, který vytváří aplikaci. Role ilustrátora se uplatní ve fázi kreslení ilustrace. Pokud programátor vyvine na základě získaných odborných znalostí o funkci zobrazené problematiky od experta ve spolupráci s grafikem aplikaci, která bude umožňovat vykreslit geometrii objektů pomocí zobrazovací techniky, která se snaží napodobit ruční kresbu, a bude se řídit podle pravidel, které by použil ilustrátor při vyhodnocení, jak daný objekt vhodně nakreslit (např. stanovení nejvhodnějšího pohledu na geometrii, určení správného umístění grafické informace do ilustrace), pak by tato aplikace částečně nahradila roli ilustrátora. Je prakticky nemožné vytvořit takovou robustní aplikaci, která by dokázala plnohodnotně zastat funkci ilustrátora. Příklad aplikace, která se snaží nahradit roli ilustrátora, je automatické umísťování popisků, kdy se stanoví kritéria, která používají ilustrátoři pro umísťování popisků, a podle těchto kritérií se snaží aplikace optimalizovat umístění popisků. Programátor takové aplikace se snaží komunikovat s ilustrátory a nebo se pokusí stanovit daná pravidla podle již existujících ilustrací podobné problematiky.

V současné době existuje množství aplikací (např. SolidWorks Motion[32], 3DVIA[1], Blender[6]), které obsahují nástroje, jenž pomohou uživateli vytvořit animaci libovolné soustavy 3D objektů. Avšak většina existujících aplikací požaduje po uživateli pro vytvoření animace soustavy 3D objektů nezanedbatelnou míru interakce, která zahrnuje například za-

dání omezení pohybu a parametrů pohybu jednotlivých objektů. Účelem postupně se vyvíjejících technických prostředků je „usnadnit uživateli život“, a proto i požadavkem na animaci pohybu soustavy mechanických součástí je minimalizovat množství potřebné interakce uživatele. Minimalizování potřebné interakce uživatele znamená, že některé údaje, které musel zadat uživatel, nyní budou automaticky vypočteny aplikací. Aplikace vypočte požadovanou informaci na základě analyzování vstupních dat (např. geometrie 3D objektu, pomocné textové informace). Uživatel tak může používat aplikační software bez hlubší znalosti problematiky, kterou se zabývá daná aplikace. Dochází k minimalizování potřebných znalostí uživatele pro práci s aplikací a takovýto software nemusí být použitelný pouze pro vzájemně spolupracující experty a grafiky v dané oblasti, ale je použitelnější pro širší spektrum lidí.

1.2 Cíle práce

Cílem této práce je analyzovat existující metody pro vizualizaci pohybu soustavy mechanických součástí v podobě animace a v podobě statického obrázku s využitím grafických informací ve formě šipek. Implementace této práce má být realizována v programovacím jazyku C++ a s využitím knihovny OpenGL. V této práci má být realizován algoritmus pro nalezení osy symetrie součástí soustavy, které mají tvar podobný válci nebo komolému kuželi. Nalezení párů součástí soustavy, které mají spolu kontakt a pro které se nemění množina součástí, s nimiž mají kontakt během vykonávání jejich pohybu, má být provedeno použitím metody detekce kolizí. Má být vytvořen graf interakce a má být vytvořena dynamická vizualizace a statická ilustrace soustavy mechanických součástí s pomocnou grafickou informací ve formě šipek, která bude znázorňovat směr pohybu pohyblivých součástí. Funkčnost této aplikace má být ověřena na 5 až 10 soustavách, které jsou složeny z 10 až 30 součástí.

1.3 Struktura práce

Následující text práce je rozdělen do 4 kapitol. Kapitola 2 popisuje analýzu existujících metod potřebných pro vizualizaci soustavy mechanických součástí pomocí animace i pomocí statického obrázku. V této kapitole je také popsán návrh principu realizovaných metod v této práci. Kapitola 3 popisuje realizaci aplikace se zaměřením se na nestandardní části řešení, použité knihovny a význačné třídy a metody realizované v této práci. V kapitole 4 jsou nejprve popsány dosažené výsledky pro stanovení vlastností součástí soustavy. Dále jsou v této kapitole uvedeny vytvořené grafy interakce a příklady vytvořených statických ilustrací analyzovaných soustav mechanických součástí. Analyzované soustavy, které jsou uvedeny v této kapitole, tvoří podmnožinu testovacích soustav. Ostatní testovací soustavy, jejich vytvořené grafy interakce a příklady vytvořených statických ilustrací se šipkami jsou uvedeny v příloze C. Kapitola 5 vyhodnocuje splnění cílů a navrhuje možná rozšíření této práce.

Kapitola 2

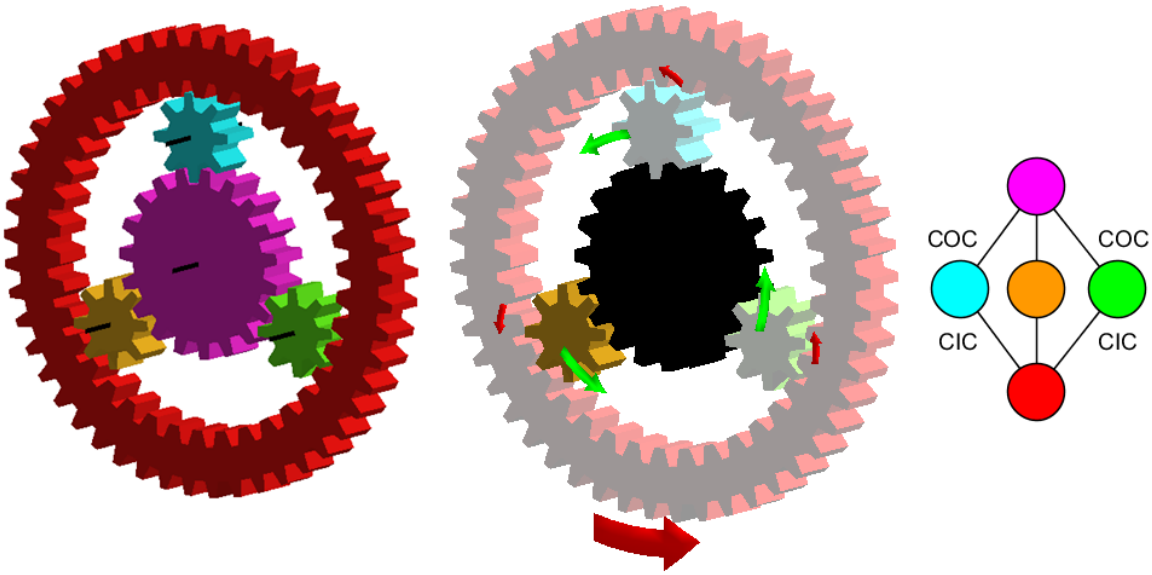
Analýza a návrh

V této kapitole je popsána analýza metod potřebných pro vizualizaci funkce soustavy mechanických součástí. Cílem je umožnit vizualizaci a přitom minimalizovat množství potřebných vstupních dat na minimum. Analýza vychází z článku Mitra a kol. [27].

V následujícím textu jsou nejprve popsány kroky potřebné pro vytvoření vizualizace a jejich vzájemná návaznost. Dále je analyzován každý z uvedených kroků zvlášť v samostatné sekci.

Určení vlastností součásti. Vizualizace pohybu soustavy mechanických součástí představuje vizualizaci pohybu jednotlivých součástí, které tvoří analyzovanou soustavu. Aby bylo možné vizualizovat pohyb součásti, je potřeba stanovit, jak se bude tato součást chovat uvnitř soustavy v okamžiku, kdy jí bude udělena kinetická energie. Chování součásti uvnitř soustavy závisí na jejích vlastnostech. Množina vlastností součásti, které je potřeba znát, závisí na podporovaných druzích objektů (např. objekt vykonávající translační pohyb, objekt vykonávající rotační pohyb) a na použité metodě pro výpočet dynamické vizualizace a statické ilustrace soustavy, která pro svoji činnost požaduje znalost příslušných vlastností součásti. Na obrázku 2.1(vlevo) je znázorněn příklad analyzované soustavy mechanických součástí se zvýrazněnými rotačními osami (černá úsečka). Přehled existujících metod pro určení vlastností objektů a popis realizované metody v této práci je uveden v podkapitole 2.1.

Nalezení součástí soustavy, které spolu mají kontakt. Aby došlo k rozpohybování soustavy mechanických součástí, je potřeba udělit alespoň jedné řídící součásti uvnitř této soustavy kinetickou energii. Jakmile bude udělena součásti libovolná nenulová velikost kinetické energie, tak se začne součást pohybovat. Pro stanovení, jak se bude kinetická energie udělená řídící součásti šířit soustavou, je potřeba určit, pro jaké dvojice součástí soustavy existuje kontakt. Pro určení, pro jaké dvojice součástí soustavy existuje kontakt, se použije technika detekce kolizí. Aby existoval kontakt dvojice součástí soustavy, tak se součásti nemusí dotýkat, ale stačí pouze, aby byly u sebe dostatečně blízko. Kontaktem dvojice součástí se rozumí taková skutečnost, že pokud zvětšíme oba objekty o stejnou definovanou hodnotu, tak bude nalezena kolize zvětšené dvojice objektů. Přehled existujících metod detekce kolizí a podrobný popis realizované metody v této práci je uveden v podkapitole 2.2.



Obrázek 2.1: (vlevo) Analýza součástí soustavy. (uprostřed) Statická ilustrace analyzované soustavy se šipkami. (vpravo) Analýza vztahů mezi součástmi soustavy. Typ spojení dvojice součástí v soustavě (vlevo) je buď válec na válci (COC) a nebo válec ve válci (CIC).

Vytvoření grafu interakce. Aby bylo možné stanovit, jaký pohyb vykonávají jednotlivé součásti soustavy po udělení kinetické energie řídící součásti, je potřeba sloučit získané informace o vlastnostech součástí soustavy a o párech součástí soustavy, které spolu mají kontakt, do grafu. Tento graf nazveme grafem interakce, protože obsahuje informace o interakci (vzájemném působení) součástí na sebe. Vzájemným působením součástí na sebe rozumíme takovou skutečnost, že pohyb jedné součásti vyvolá pohyb součástí soustavy, které mají kontakt s touto součástí. Graf interakce je neorientovaným grafem, jehož vrcholy představují součásti, které obsahují informaci o svých vlastnostech, a hrany znázorňují existující kontakt příslušné dvojice součástí. Příklad grafu interakce pro soustavu součástí (viz obrázek 2.1(vlevo)) je znázorněn na obrázku 2.1(vpravo), kde barvy uzlů grafu interakce odpovídají barvám součástí, které představují. Popis hrany grafu představuje typ spojení dvojice součástí, který je podrobně popsán v podkapitole 2.3. Princip vytvoření grafu interakce, který je realizován v této práci, je uveden v podkapitole 2.3.

Dynamická vizualizace soustavy mechanických součástí. Dynamickou vizualizací soustavy mechanických součástí rozumíme animaci pohybu součástí, které tvoří analyzovanou soustavu mechanických součástí. Aby bylo možné znázornit pohyb součásti, je potřeba nejprve stanovit parametry jejího pohybu, kterými jsou směr, rychlost a osa rotace součásti. Parametry pohybu součásti soustavy jsou stanoveny na základě znalosti řídící součásti a grafu interakce analyzované soustavy. Princip metody pro stanovení parametrů pohybu součástí soustavy, který je realizován v této práci, je uveden v podkapitole 2.4.

Statická ilustrace soustavy součástí s grafickými informacemi. Statickou ilustrací soustavy mechanických součástí rozumíme vytvoření obrázku, na kterém bude znázorněna analyzovaná soustava mechanických součástí z takového pohledu, na kterém je co nejlépe zřetelná její vysvětlovaná funkce. Grafické informace mají za úkol pomoci lépe porozumět vytvořené ilustraci a mohou být ve formě např. šipek. Pro vytvoření statické ilustrace se šipkami je potřeba znát parametry pohybu jednotlivých součástí analyzované soustavy, které byly vypočteny pro vytvoření dynamické vizualizace soustavy mechanických součástí, a graf interakce analyzované soustavy. Aby bylo možné vytvořit statickou ilustraci soustavy se šipkami, je potřeba znát směr, jakým se pohybují součásti soustavy, který je znázorněn ve statické ilustraci šipkou. Pro nalezení nejvhodnějšího umístění šipek ve statické ilustraci je potřeba znát graf interakce analyzované soustavy. Příklad statické ilustrace si šipkami soustavy mechanických součástí je znázorněn na obrázku 2.1(uprostřed), kde zvýrazněná žlutá součást představuje řídící součást a černá součást označuje nepohyblivou součást. Na obrázku představuje zelená šipka rotaci součásti okolo rotační osy nepohyblivé součásti a červená šipka znázorňuje rotaci součásti okolo její osy rotace. Přehled existujících metod pro vytvoření statické ilustrace s pomocnou grafickou informací a princip metody, která je realizována v této práci, je uveden v podkapitole 2.5.

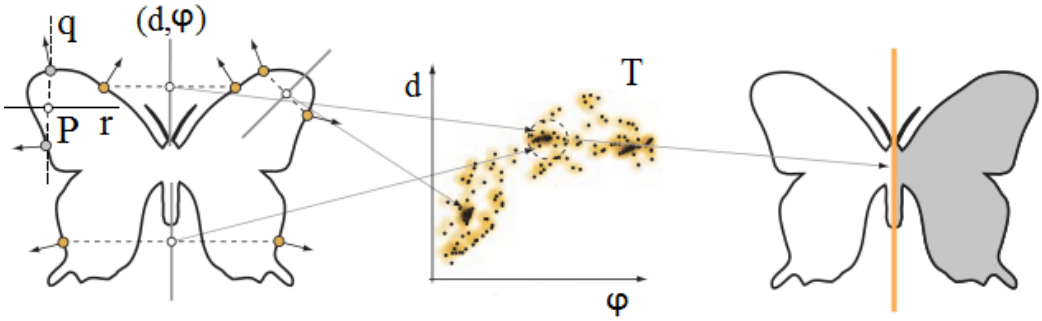
2.1 Určení vlastností součásti

V této práci jsou podporovány pouze rotační součásti, které mají tvar válce (hřídel, ozubené kolo, spirálové ozubené soukolí) nebo komolého kužele (kuželové ozubené soukolí). Hřídel představuje podlouhlou válcovou součást, na které jsou nasazeny ozubené součásti, které se otáčejí s hřídelí okolo její osy rotace. Tento požadavek plyne ze zadání práce, že mají být podporovány 3D objekty, které mají tvar podobný válci nebo komolému kuželi. Omezení na tyto dva tvary 3D objektů je z důvodu oblasti použití této práce, kterou je strojírenský průmysl.

V ideálním případě bychom chtěli rozpoznat libovolně složitý 3D objekt (např. vačku, ozubnici) a určit jeho vlastnosti pouze ze znalosti geometrie. Obecně je však tento problém příliš složitý a jde nad rámec diplomové práce. Množina vlastností 3D objektu je závislá na jeho tvaru. Pro určení různých parametrů objektu existuje množství metod. Pro vytvoření dynamické vizualizace soustavy je potřeba znát osu rotace, počet zubů a tvar součásti. Pro vytvoření statické ilustrace se šipkami je potřeba znát osu rotace, poloměr podstav, tvar a výšku součásti. Pokud je součást tvaru válce, pak výškou součásti rozumíme výšku válce. Pokud je součást tvaru komolého kužele, pak výškou součásti rozumíme výšku komolého kužele. Pro součásti podporované v této práci platí, že jejich osa rotace je totožná s jejich osou symetrie. V této práci bude stanovena osa rotace, tvar, výška, poloměry podstav a počet zubů každé součásti analyzované soustavy na základě analýzy její geometrie.

V této podkapitole budou nejprve popsány principy dvou analyzovaných metod a poté bude popsán princip realizované metody v této práci. První analyzovanou metodou je metoda, která nalezne osu symetrie objektu. Druhou analyzovanou metodou je metoda, která nalezne rotační osu 3D objektu na základě nalezení jeho kruhových otvorů či výstupků, ve kterých mohou rotovat jiné objekty.

Obrázek 2.2: Znázornění použití transformace (vlevo) zrcadlení, (uprostřed) zrcadlení, rotace a posun, (vpravo) změna měřítka, rotace a posun pro detekci symetrických částí objektu. Texty obrázku byly přeloženy do češtiny autorkou práce. Zdroj obrázku [26].



Obrázek 2.3: (vlevo) Dvojice bodů, které leží na kontuře objektu, a jejich osa zrcadlení. (uprostřed) Transformační prostor T (vodorovně: velikost výškově-azimutálního úhlu, svisle: 1/2 vzdálenosti dvojice bodů). (vpravo) Hlavní osa zrcadlení je největší shluk v prostoru T. Texty obrázku byly přeloženy do češtiny autorkou práce. Zdroj obrázku [26].

2.1.1 Nalezení osy symetrie objektu

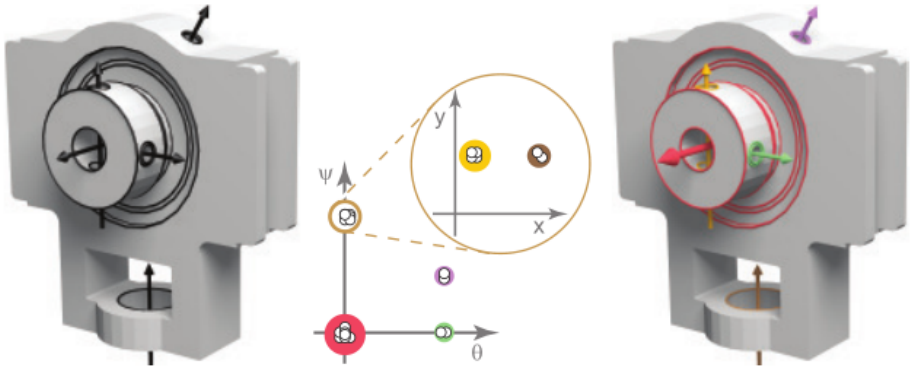
Symetrií 3D objektu rozumíme neměnnost vůči transformaci zrcadlení. Osa symetrie pak udává střed zrcadlení.

Princip této metody prezentují Mitra a kol. [26]. Vstupem algoritmu bude množina vstupních bodů, které leží na povrchu 3D objektu. Popisovaný princip nalezení osy symetrie je znázorněn na obrázku 2.3 pro objekt, jehož kontury připomínají tvar motýla. Pro každý pár vstupních bodů se vytvoří přímka q, která prochází oběma body. Pomocí Euklidovy vzdálenosti se určí poloviční vzdálenost každého páru. Bod, který leží na přímce q v poloviční vzdálenosti mezi dvojicí bodů, bude označen jako bod P. Vytvoří se přímka r, která je kolmá na přímku q a která prochází bodem P. Přímka r představuje potencionální osu symetrie motýla. Popsaný princip pro stanovení potencionálních os symetrie součásti na základě analyzování množiny vstupních bodů je znázorněn na obrázku 2.3(a). Nalezená osa symetrie (přímka r) se transformuje pomocí transformační funkce do transformačního prostoru, kde osa symetrie odpovídá bodu v tomto prostoru. Transformační prostor je ortogonálním prostorem, protože jeho bázové vektory jsou na sebe kolmé a jsou normalizovány. Na vodorovné souřadnicové ose transformačního prostoru je vynášena velikost úhlu, která je rovna velikosti výškově-azimutálního úhlu. Na vertikální osu transformačního prostoru je vynášena Euklidova vzdálenost definovaného páru bodů. Vytvoří se shluky bodů v transformačním prostoru (viz obrázek 2.3(b)). Pro každý shluk se vytvoří reprezentant. Reprezentantem bude například taková nalezená osa, která představuje bod v transformačním prostoru, který je těžištěm shluku. Mohutnost shluků bodů vypovídá o významnosti dané osy symetrie (viz obrázek 2.3(c). Hlavní osou symetrie 3D objektu je reprezentant shluku s největší mohutností. Výhodou této metody je, že dokáže detekovat i částečné symetrie objektu. Při použití rozdílné transformační funkce a rozdílného transformačního prostoru lze tuto metodu použít i pro detekci sobě podobných částí 3D objektu (viz obrázek 2.2).

2.1.2 Nalezení rotační osy 3D objektu

Mitra a kol. [27] prezentují princip metody, která určí rotační osu objektu a nalezne kruhové otvory/výstupky objektu ve/na kterých mohou rotovat jiné objekty. Požadovaným vstupem této metody je povrchová reprezentace 3D modelu, která je typicky tvořena trojúhelníkovou sítí. Pomocí hladového algoritmu dojde k nalezení takových oblastí objektu, které jsou odděleny ostrými hranami. Nechť hrana e_k 3D modelu spojuje dvojici stěn f_i, f_j, které tvoří povrch modelu, pak o stěnách f_i, f_j lze říci, že jsou sousedními stěnami. Jako ostrá hrana je označena taková hrana, která spojuje takové dvě sousední stěny f_i,f_j, pro které platí, že velikost úhlu, který svírají jejich normálové vektory, je větší nežli definovaná mez pro ostrou hranu (např. 45°).

Pro stanovení, jaké ostré hrany ohraničují kruhové oblasti součásti, je použita metoda nejmenších čtverců. Dojde ke stanovení poloměru, středu a směru normály každé nalezené kruhové oblasti. Normálový vektor kruhové oblasti součásti je totožný se směrovým vektorem osy, která prochází středem kruhového otvoru, který je soustředný s kruhovou oblastí součásti. Seskupí se osy kruhových oblastí, které mají přibližně stejný směr. Seskupení os se provede na základě parametrizace os v prostoru přímek podle velikosti jejich výškových a azimutálních úhlů.



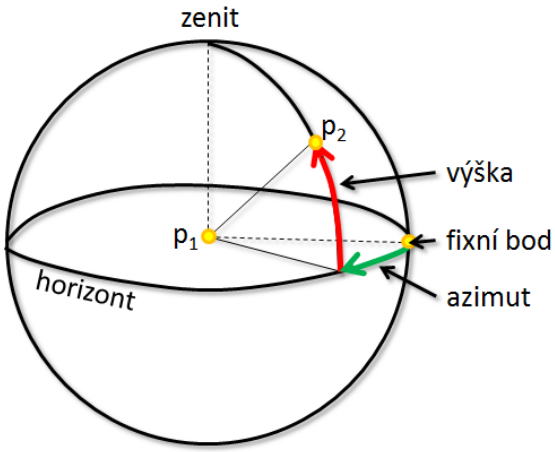
Obrázek 2.4: Obrázek znázorňuje princip nalezení osy rotace 3D objektu (vlevo). Pomocí metody nejmenších čtverců se detekují kruhové oblasti (zvýrazněné kružnice na obrázku vlevo i vpravo). Seskupí se osy kruhových oblastí podle jejich směru (uprostřed). Rozdělí se vzniklé shluky podle projekcí středů kruhových oblastí (hnědá a žlutá osa). Mohutnost shluku odpovídá tloušťce osy (vpravo). Zdroj obrázku [27].

Princip, jak se určí výškový a azimutální úhel, je znázorněn na obrázku 2.5. Jednotlivé přímky jsou definovány pomocí dvou bodů p_1 a p_2, kde bod p_1 je totožný se středem koule a bod p_2 leží na povrchu koule (viz obrázek 2.5). Pozice bodu na povrchu koule je určena pomocí výškového a azimutálního úhlu. Hodnota výškového úhlu je v rozsahu od 0° do 90°. Hodnota azimutálního úhlu je v rozsahu od 0° do 360°. Pomocí tohoto souřadného systému lze popsat směr přímky (vektor). Definuje se malý úhel přibližně 4°, který určuje, jaké přímky budou společně seskupeny. Osy vyskytující se ve stejném shluku jsou téměř rovnoběžné. Bude určeno těžiště pro každý shluk, které bude představovat jednu přímku jako reprezentanta dané skupiny. Vzniklé shluky budou dále rozděleny podle kolmých projekcí středů kruhových oblastí na rovinu, která je kolmá na reprezentanta. Tímto postupem dojde k seskupení os, které jsou totožné. Osou rotace součásti bude taková osa, která tvoří shluk s největší mohutností. Uvedený princip je zobrazen na obrázku 2.4.

2.1.3 Realizovaná metoda

Metoda realizovaná v této práci je nejpodobnější popsané metodě stanovující rotační osu 3D objektu. Tyto dvě metody mají společné, že využívají znalosti ostrých hran objektu. Detekce ostrých hran není v realizované metodě prováděna nad trojúhelníkovou sítí, která tvoří povrch modelu, ale je prováděna nad konvexní obálkou součásti.

Pro vypočtení osy rotace, poloměru podstav a tvaru součásti je potřeba vypočítat lokální souřadnicový systém součásti, stanovit konvexní obálku součásti, nalézt extrémní vrcholy konvexní obálky a vytvořit shluky nalezených extrémních vrcholů. Pro stanovení rovin, ve kterých leží podstavy součásti, je potřeba znát extrémní vrcholy konvexní obálky a lokální souřadnicový systém součásti. Ze znalosti rovin, ve kterých leží podstavy součásti, se stanoví výška součásti. Znalost extrémních vrcholů konvexní obálky je potřeba také pro stanovení počtu zubů součásti.



Obrázek 2.5: Střed souřadnicového systému je totožný se středem koule označeným jako p_1. Bod na povrchu koule, který je označen jako p_2, je určen výškovým a azimutálním úhlem. Přímka prochází středem koule a bodem na povrchu koule.

Pro určení lokálního souřadnicového systému součásti se použije metoda analýzy hlavních komponent. Lokální souřadnicový systém součásti je tvořen počátkem soustavy souřadnic a bázovými vektory, které jsou vzájemně na sebe kolmé. Lze říci, že lokální souřadnicový systém je shodný s kartézskou soustavou souřadnic. Osou rotace součásti je vždy jedna ze souřadnicových os jejího lokálního souřadnicového systému.

Pro stanovení, jaká osa lokálního souřadnicového systému je osou rotace součásti, je potřeba stanovit konvexní obálku součásti a nalézt ostré hrany ve vytvořené obálce. Vrcholy součásti, které budou ležet na těchto ostrých hranách, budou hledanými extrémními vrcholy.

Pro každou osu lokálního souřadnicového systému součásti se vytvoří shluky nad seřazenou posloupností vypočtených vzdáleností extrémních vrcholů konvexní obálky k dané ose lokálního souřadnicového systému. Podle mohutnosti vzniklých shluků se stanoví osa rotace součásti.

V následujícím textu jsou podrobně analyzovány techniky použité v realizované metodě.

Určení lokálního souřadnicového systému součásti

Pro nalezení lokálního souřadnicového systému součásti se použije princip metody analýzy hlavních komponent, jejíž zkratka je PCA (z angl. principal component analysis). Jolliffe[24] prezentuje princip metody PCA, která určí osy (bázové vektory) a počátek lokálního souřadnicového systému součásti. Počátek lokálního souřadnicového systému udává umístění objektu a odpovídá pozici těžiště součásti. Směrové vektory os lokálního souřadnicového systému udávají natočení součásti v prostoru. PCA se používá pro redukci dimenze vstupní množiny dat za účelem zjednodušení těchto dat a je založena na základním principu nalezení takové projekce vstupní množiny dat, která nejlépe reprezentuje data ve smyslu nejmenších čtverců. PCA je podobná lineární regresi s použitím metody nejmenších čtverců. V této práci jsou vstupní množinou dat vrcholy, které leží na povrchu objektu ve 3D. Úkolem této

metody je nalézt takový vektor, který prochází počátkem lokálního souřadnicového systému součásti tak, že součet kvadrátů vzdáleností mezi body ze vstupní množiny dat a jejich příslušnými projekcemi na vypočtený bázový vektor je tak malá, jak je jen možné. Počátek lokálního souřadnicového systému součásti se určí jako aritmetický průměr vstupní množiny dat. Pro 3D data se provádí výpočet aritmetického průměru zvlášť pro každou souřadnicovou osu x, y a z. Standardní odchylka množiny dat udává rozptyl dat. Jedná se o průměrnou vzdálenost od průměru množiny dat. Standardní odchylka se vypočte podle následujícího vzorce:

$$std = \sqrt{\frac{\sum_{i=1}^{n}(X_i - X)^2}{n-1}}, \tag{2.1}$$

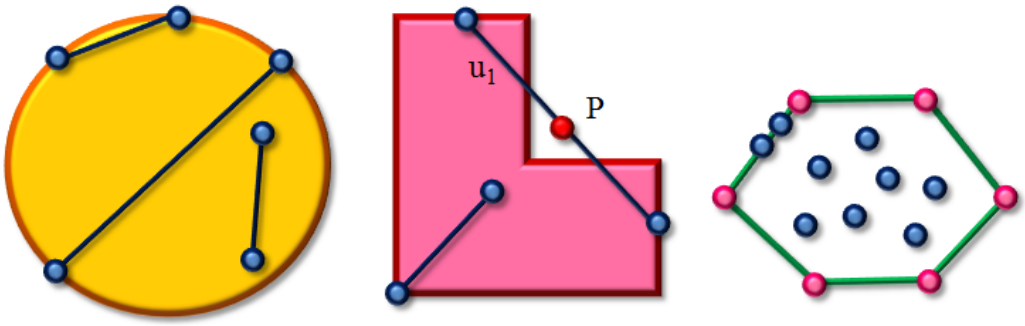
kde X_i je i-tý prvek vstupní množiny dat, X je průměr vstupní množiny dat a n je mohutnost vstupní množiny. Hodnota variance se rovná druhé mocnině standardní odchylky. Standardní odchylka i variance jsou jednodimenzionální hodnoty. Pokud by bylo potřeba znát vztah mezi dimenzemi vstupní množiny dat, jejíž data by byla vícerozměrná, pak by bylo potřeba použít kovarianci. Kovariance se měří vždy mezi dvojicí dimenzí. Kovariance se spočítá pomocí následujícího vzorce:

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - X)(Y_i - Y)}{(n-1)}, \tag{2.2}$$

kde X_i je i-tý prvek jedné dimenze vstupní množiny dat, X je její aritmetický průměr a obdobný je význam symbolů Y_i, Y pro druhou dimenzi té samé vstupní množiny dat. Pokud je požadováno znát vztah mezi všemi dimenzemi 3D dat, pak je potřeba určit kovarianci $cov(x,y)$, $cov(x,z)$ a $cov(y,z)$, kde $cov(x,y)$ stanoví vztah pro vstupní množinu 3D dat v ose x a y. Kovarianční matice pro 3D množinu dat má tři řádky a tři sloupce a její struktura vypadá následovně:

$$\mathbf{K} = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix} \tag{2.3}$$

Kovarianční matice je symetrická podle hlavní diagonály, což znamená, že například hodnota $cov(y,x)$ je totožná s hodnotou $cov(x,y)$. Nechť $\mathbf{K}$ je kovarianční matice, která je čtvercovou maticí reálných nebo komplexních čísel. Olšák [29] definuje pojem vlastní číslo tak, že číslo $\lambda \in \mathbb{C}$ se nazývá vlastním číslem matice $\mathbf{K}$, pokud existuje vektor $\mathbf{x} \in \mathbb{C}^n$, $\mathbf{x} \neq \mathbf{o}$ takový, že $\mathbf{K}\mathbf{x}^\mathbf{T} = \lambda\mathbf{x}^\mathbf{T}$. Symbol $\mathbf{o}$ značí nulový vektor v lineárním prostoru. Olšák [29] pak definuje pojem vlastní vektor tak, že vektor $\mathbf{x}$, který splňuje uvedenou rovnost, se nazývá *vlastní vektor* matice $\mathbf{K}$ *příslušný vlastnímu číslu* λ. Vlastní číslo udává délku vlastního vektoru. Vlastní vektory tvoří ortogonální bázi lokálního souřadnicového systému součásti. Z definice vlastního čísla je známo, že vlastní vektor musí být nenulový. Číslo λ bude podle Olšáka [29] vlastním číslem matice $\mathbf{K}$, když homogenní soustava s maticí $\mathbf{K}$ - $\lambda\mathbf{E}$ bude mít nenulové řešení, které odpovídá vlastnímu vektoru příslušnému vlastnímu číslu. Matice $\mathbf{E}$ značí jednotkovou matici. Jednotková matice je taková čtvercová matice, pro jejíž prvky $e_{i,j}$ platí: $e_{i,j} = 0$ pro $i \neq j$ a $e_{i,j} = 1$ pro $i = j$. Vlastní čísla matice $\mathbf{K}$ se vypočtou z rovnice $\det(\mathbf{K} - \lambda\,\mathbf{E}) = 0$, kde $\det(\mathbf{K} - \lambda\mathbf{E})$ se nazývá *charakteristickým polynomem* matice $\mathbf{K}$. Počet vlastních čísel matice $\mathbf{K}$ je roven počtu řádků této matice. Stupeň charakteristického polynomu matice $\mathbf{K}$ je shodný s počtem řádků této matice. Vlastní čísla matice $\mathbf{K}$ se naleznou jako kořeny charakteristického polynomu této matice.



Obrázek 2.6: (vlevo) Konvexní množina. (uprostřed) Nekonvexní množina. (vpravo) Konvexní obálka a její extrémní vrcholy (růžová barva), které jsou koncovými body hran konvexní obálky (zelená barva).

Výpočet konvexní obálky součásti a nalezení jejích extrémních vrcholů

Na obrázku 2.6 je zobrazen příklad konvexní množiny, nekonvexní množiny a konvexní obálky. Konvexní množina je taková podmnožina Euklidovského prostoru, pro kterou platí, že přímka, která spojuje libovolné dva vrcholy množiny, leží v této množině. Množina na obrázku 2.6(b), není konvexní množinou, protože například bod P, který leží na úsečce u_1, není obsažen v této množině. Na obrázku 2.6(c) je zobrazen příklad konvexní obálky množiny bodů. Konvexní obálka množiny S je podle Preparaty a kol. [31] taková nejmenší množina, která obsahuje S. Konvexní obálka představuje například ve 2D takový nejmenší konvexní mnohoúhelník, který obsahuje všechny body množiny. Vrcholy konvexního mnohoúhelníku jsou označeny jako extrémní vrcholy (viz obr. 2.6(c) – růžově označené vrcholy). Pro případ 2D extrémním vrcholem rozumíme takový vrchol, který je koncovým bodem úsečky. Konvexní obálka množiny bodů, které jsou zobrazeny na obrázku 2.6(a), je shodná s hranicemi této konvexní množiny (viz obrázek 2.6(a) – oranžová kružnice). Konvexní obálkou objektu ve 3D je konvexní mnohostěn. Extrémní vrchol konvexní obálky ve 3D je vrchol, který je jedním z vrcholů konvexního mnohostěnu. Pokud by byly přidány do konvexní obálky i vrcholy, které by měly shodné souřadnice s některým již přidaným vrcholem nebo by ležely na stěně konvexní obálky, pak by tato konvexní obálka neobsahovala pouze extrémní vrcholy.

Popis algoritmů pro vytvoření konvexní obálky je uveden v příloze B. V této práci byla pro vytvoření konvexní obálky mechanické součásti použita metoda Quick hull (viz B.2). Získaná konvexní obálka součásti obsahuje nejen extrémní vrcholy ale i vrcholy, které leží na stěnách konvexní obálky a které nejsou extrémními vrcholy. Z důvodu získání správné množiny vrcholů pro realizovanou metodu stanovení vlastností součásti je potřeba zajistit, že získané vrcholy z konvexní obálky budou jejími extrémními vrcholy, které leží na jejích ostrých hranách. Proto bylo potřeba dodatečně určit, jaké hrany konvexní obálky jsou ostrými hranami. Pokud je velikost úhlu, který svírají stěny konvexní obálky, které jsou spojeny hranou, větší nežli je mez pro ostrou hranu (např. 40°), pak je tato hrana prohlášena za ostrou hranu a vrcholy, které leží na této ostré hraně, jsou extrémními vrcholy konvexní obálky.

Výška součásti je rovna vzdálenosti roviny π_r a π_g, ve kterých leží podstavy součásti (viz obrázek 2.8(a)). Počet zubů součásti se stanoví jako podíl počtu extrémních vrcholů součásti,

které leží na jedné z podstav součásti, hodnotou, která je pevně definována uživatelem a která stanovuje, kolik extrémních vrcholů tvoří jeden zub. Pokud je analyzovanou součástí součást, která má vnitřní a vnější zuby, pak počet vnějších zubů se zjistí pomocí popsaného principu a počet vnitřních zubů je roven podílu počtu vrcholů, které mají minimální vzdálenost k ose rotace součásti ze všech vrcholů, které tvoří povrch součásti, počtem vrcholů, které mají tvořit jeden vnitřní zub součásti. Počet vnitřních zubů součásti nelze spočítat ze znalosti počtu extrémních vrcholů konvexní obálky součásti, protože tyto zuby neleží na konvexním mnohostěnu, který tvoří konvexní obálku součásti ve 3D.

Vytvoření shluků

Byly stanoveny osy lokálního souřadnicového systému součásti a extrémní vrcholy konvexní obálky. Pro určení, jaká osa lokálního souřadnicového systému součásti je osou rotace, je potřeba vypočítat vzdálenost extrémních vrcholů konvexní obálky k jednotlivým souřadnicovým osám a vytvořit shluky vypočtených vzdáleností pro každou osu zvlášť. Pro určení osy rotace je nutné uvažovat pouze extrémní vrcholy konvexní obálky analyzované mechanické součásti, které tvoří podmnožinu množiny vrcholů ležících na jejím povrchu. Pokud by nebyly uvažovány pouze extrémní vrcholy konvexní obálky, pak by dle dále popsaného postupu mohlo dojít ke špatné detekci rotační osy vlivem vrcholů, které by chybně zvyšovaly mohutnost vytvořených shluků.

Pro každou z os lokálního souřadnicového systému se vypočte vzdálenost všech extrémních vrcholů konvexní obálky součásti od této osy. Pro každou z os se vypočtené vzdálenosti seřadí do neklesající posloupnosti. Pro jednotlivé osy se vytvoří shluky tak, že pokud i–tá a i–1 hodnota v seřazené posloupnosti vzdáleností překročí definovanou mez, pak se vytvoří nový shluk a i–tá hodnota se přidá do nově vzniklého shluku. Je požadováno, aby pouze pro jednu osu lokálního souřadnicového systému existoval jeden nebo dva shluky. Ostatní osy lokálního souřadnicového systému musí mít více jak dva shluky. Osa, pro kterou budou existovat jeden nebo dva shluky bude osou rotace součásti. Algoritmus 1 popisuje postup vytvoření shluků na základě stanovené hodnoty meze, která se určuje v algoritmu 2.

V algoritmu 2 se uchovávají hodnoty proměnných *mezStará*, *mezNová* a *krok* z posledního průchodu. Proměnná MIN se získá jako minimum rozdílu dvou sousedních vzdáleností přes všechny souřadnicové osy a všechny extrémní vrcholy. Obdobně je získána hodnota proměnné MAX. Hledání hodnoty meze odpovídá principu algoritmu binárního vyhledávání.

Pro podporované tvary součástí v této aplikaci, jimiž je válec a komolý kužel, prochází osa rotace množinou extrémních vrcholů konvexní obálky tak, že vrcholy, které leží v nejmohutnějších shlucích, tvoří soustředně promítnuté kružnice do roviny, která je kolmá na osu rotace. Na obrázku 2.7 jsou zobrazeny histogramy mohutností shluků pro osy lokálního souřadnicového systému součásti, která má tvar komolého kužele a válce. Na obrázku 2.7 je též uvedena mez shluku, pro kterou byly vytvořeny uvedené histogramy, která určuje, při jakém rozdílu sousedních hodnot vzdáleností v seřazené posloupnosti dojde k vytvoření nového shluku. Plnou čarou je zobrazena rotační osa součásti.

Pro správnou funkci detekce osy rotace a detekce tvaru součásti je potřeba uvažovat pouze takové 3D modely mechanických součástí, jejichž extrémní vrcholy konvexní obálky leží na jejich podstavách. Na obrázku 2.8(a) jsou zobrazeny roviny π_r a π_g, ve kterých leží extrémní vrcholy konvexní obálky. Obě dvě roviny jsou rovnoběžné s rovinou π_y, jejíž normálový vektor

Algoritmus 1 Vytvoř shluky a najdi osu rotace součásti

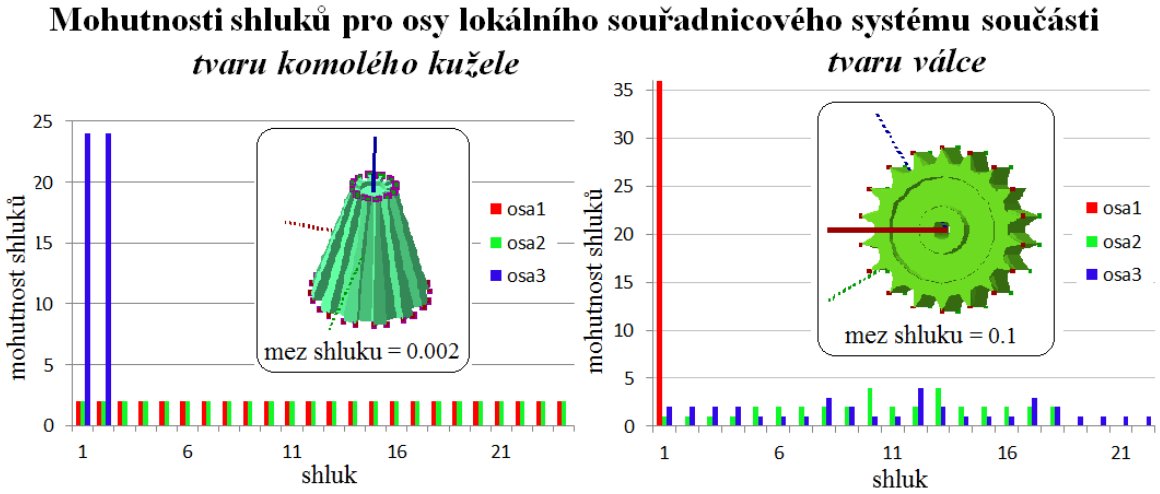
Vstup: osy součásti a extrémní vrcholy konvexní obálky (CH)
Výstup: vytvořené shluky a osa rotace součásti

```
1: Vypočti vzdálenosti extrémních vrcholů CH k osám lokál. souřad. systému.
2: Seřaď vzdálenosti do neklesající posloupnosti pro jednotlivé souřadnicové osy.
3: Mez shluku = Urči mez shluku(INIT).
4: Pro stanovenou mez vytvoř shluky pro jednotlivé osy.
5: Zkontroluj počet shluků pro jednotlivé souřadnicové osy.
6: if Pokud existují 1 nebo 2 shluky pouze pro 1 osu. then
7:    Osa obsahující 1 nebo 2 shluky je osou rotace.
8:    return  Konec algoritmu.
9: else
10:   Hledej novou hodnotu meze.
11:   if Existuje pro více jak 1 osu 1 nebo 2 shluky. then
12:      Zmenši hodnotu meze na hodnotu = Urči mez shluku(B1).
13:      go to 4
14:   end if
15:   if Neexistuje ani pro 1 osu pouze 1 či 2 biny. then
16:      Zvětši hodnotu meze na hodnotu = Urči mez shluku(B2).
17:      go to 4
18:   end if
19: end if
```

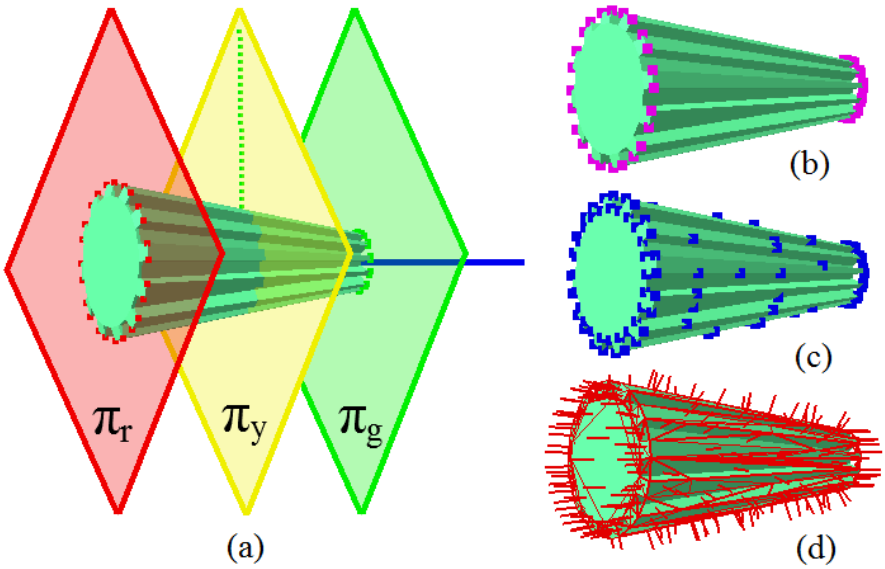
Algoritmus 2 Urči mez shluku(stav)

Vstup: stav (init/ B1 zmenši/ B2 zvětši)
Výstup: mezNová

```
1: if stav == INIT then
2:    mezStará = MIN
3:    mezNová = MIN
4:    krok = (MIN+MAX)/2
5: end if
6: if stav == B1 then
7:    krok /= 2
8:    mezNová = mezStará + krok
9: end if
10: if stav == B2 then
11:   mezStará = mezNová
12:   mezNová = mezStará + krok
13: end if
14: return  mezNová
```



Obrázek 2.7: (vlevo) Histogram pro součást tvaru komolého kužele, který má pro osu rotace dva shluky. (vpravo) Histogram pro součást tvaru válce, který má pro osu rotace jeden shluk. Osa rotace znázorněna plnou čarou.



Obrázek 2.8: (a) V rovinách π_g, π_r leží podstavy součásti. (b) Extrémní vrcholy konvexní obálky. (c) Vrcholy konvexní obálky. (d) Hrany a normály stěn konvexní obálky.

je shodný s osou rotace součásti, a ve které leží počátek lokálního souřadnicového systému součásti. Na obrázku 2.8(b) jsou zobrazeny extrémní vrcholy konvexní obálky, které leží na podstavách součásti. Na obrázku 2.8(c) vidíme vrcholy na obálce, které obsahují jak extrémní vrcholy konvexní obálky, tak i některé vrcholy na povrchu součásti, které nejsou extrémní. Pro účel správné detekce tvaru součásti a osy rotace je potřeba eliminovat tyto vrcholy pouze na extrémní vrcholy. Na obrázku 2.8(d) jsou zobrazeny normály stěn konvexní obálky, které slouží pro stanovení extrémních vrcholů konvexní obálky.

Ze znalosti rotační osy součásti se vypočte poloměr jejích podstav jako aritmetický průměr vypočtených vzdáleností, které jsou uloženy v daném shluku. Tvar součásti se stanoví na základě znalosti počtu shluků pro její rotační osu. Pokud existuje pouze jeden shluk pro rotační osu, pak je tvar součásti válec. V opačném případě je potřeba zjistit procentuální rozdíl poloměrů jejích podstav. Pokud se poloměry podstav liší minimálně o definovanou mez (např. 10%), pak je tvar součásti komolý kužel. V opačném případě je tvarem součásti opět válec.

Omezení realizované metody pro stanovení vlastností součásti

Jak již bylo řečeno, pro správnou funkci detekce osy rotace a detekce tvaru součásti je potřeba uvažovat pouze takové 3D modely mechanických součástí, jejichž extrémní vrcholy konvexní obálky leží na jejich podstavách a jejichž osa rotace je totožná s jejich osou symetrie.

Pro správné určení počtu zubů součásti je potřeba zadat správný počet extrémních vrcholů, které tvoří jeden zub součásti.

Pro případ součásti s vnitřními a vnějšími zuby a stanovení jejího počtu vnitřních zubů je kvůli požadavku na stejný počet vrcholů, které tvoří jeden zub (vnější a vnitřní), potřeba uvažovat při vytváření 3D modelu součásti, že vrcholy, které leží na povrchu objektu a které budou mít minimální vzdálenost od osy rotace součásti, musí být pouze na ostrých hranách součásti.

2.2 Nalezení součástí soustavy, které spolu mají kontakt

Pro vytvoření dynamické vizualizace a statické ilustrace je potřeba znát, jaké parametry animace bude mít každá součást soustavy. Parametry animace součástí soustavy závisí na vlastnostech součástí soustavy, které popisují, jak se součást uvnitř soustavy chová, a na znalosti párů součástí soustavy, které mají spolu kontakt. Pro určení, jak bude docházet k šíření kinetické energie udělené řídící součásti analyzovanou soustavou, je potřeba stanovit, pro jaké dvojice součástí soustavy existuje kontakt. Součásti spolu reálně nekolidují, ale pokud by došlo k pohybu řídící součásti a řízená součást by nezměnila svojí polohu, pak by došlo ke kolizi součástí. Dvojice součástí má spolu kontakt, pokud se dotýkají a nebo pokud jsou dostatečně blízko sebe. Stanovení, zda existuje kontakt dvojice součástí, je těžším úkolem nežli samotné stanovení, zda existuje kolize dvojice objektů, protože je potřeba vypočítat minimální vzdálenost mezi objekty. V této práci na místo vypočtení minimální vzdálenosti mezi objekty nejprve dojde k extrudování objektů o stejnou hodnotu a až pak dojde ke stanovení, zda objekty kolidují. Pokud existuje kolize extrudované dvojice součástí soustavy, pak existuje kontakt analyzované dvojice součástí. Hodnotu extrudování objektů soustavy pro

stanovení, zda existuje kontakt součástí, lze vypočítat například jako procentuální hodnotu délky diagonály osově zarovnané obálky analyzované soustavy.

Existují obecně dva základní typy dotazů pro detekci kolizí dvou objektů. Prvním typem dotazu je, že je požadováno pouze stanovit, zda existuje či neexistuje kolize příslušných dvou objektů. Druhým typem dotazu je, že je požadováno nalézt průsečíky dvojice testovaných objektů, pokud existuje jejich kolize.

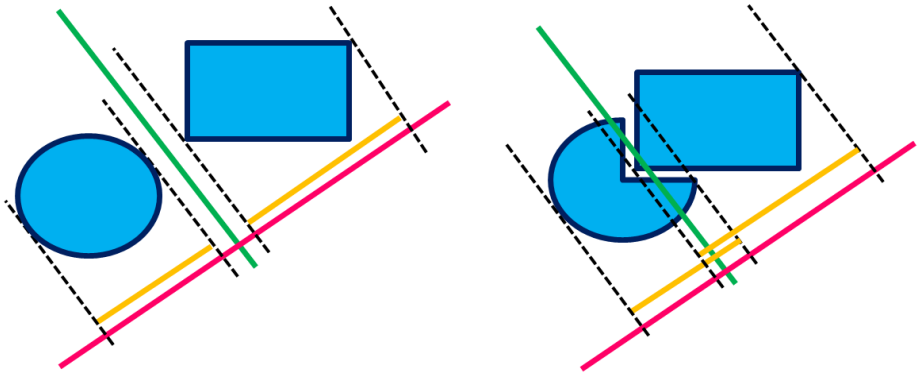
Pro detekci kolizí součástí existuje řada diametrálně odlišných technik. V následujícím přehledu některých metod, které se zabývají touto problematikou, se podíváme na metody SAT, BVH, Minkowského součet a detekci kolizí založenou na voxelizaci. V této práci je realizována metoda detekce kolizí, která je založena na voxelizaci objektů na grafickém procesoru.

2.2.1 SAT

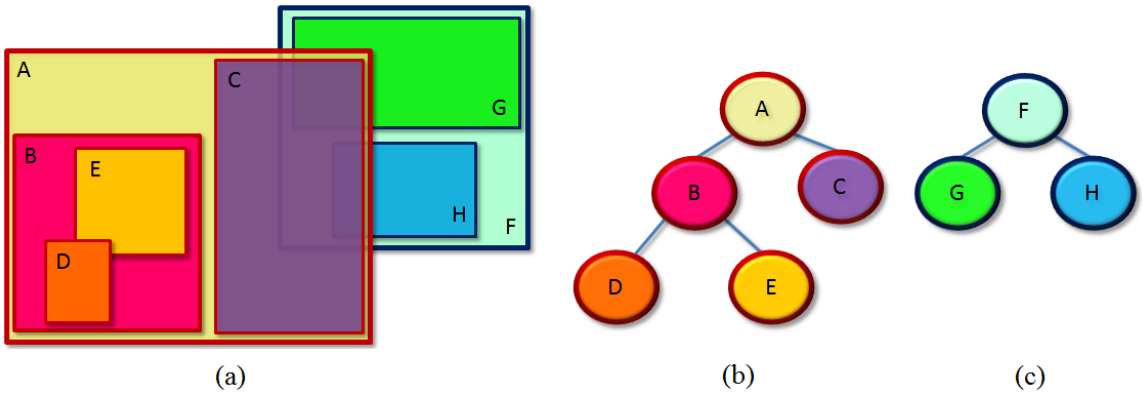
Gottschalk a kol. [21] prezentují princip metody SAT (z angl. separating axis theorem), kterou lze použít pouze v případě, že je dvojice testovaných objektů konvexní. SAT určí separační přímku (2D), která vytvoří takové dvě poloroviny, že v jedné polorovině leží pouze jeden objekt a v druhé polorovině leží druhý objekt. Vytvoří se přímka kolmá na separační přímku, která se nazývá separační osa. Na tuto přímku se vynesou kolmé projekce vrcholů testovaných dvou objektů. Každý objekt tak vytvoří na této separační ose interval, ve kterém se vyskytuje daný objekt. Maximální potřebný počet vytvořených separačních os pro stanovení, zda se dva konvexní objekty protínají, je závislý na typu použité obálky. Například pro test dvou OBB (z angl. oriented bounding box, který má libovolně orientované osy) obálek bude potřeba určit 15 ($3+3$ normál stěn, $3 \cdot 3$ vektorových součinů hran) separačních os. Pokud bude nalezena alespoň jedna separační osa, na které nedojde k průniku těchto dvou intervalů, pak se OBB obálky neprotínají. Na obrázku 2.9 je uveden příklad aplikace této metody pro 2.9(vlevo) dva konvexní objekty a pro 2.9(vpravo) alespoň jeden nekonvexní objekt. Pokud je alespoň jeden objekt nekonvexní, pak si můžeme na obrázku všimnout, že i když se objekty neprotínají, tak nelze nalézt žádnou separační přímku, která by oddělila objekty tak, že by v každé polorovině ležel pouze jeden objekt. Z tohoto důvodu by docházelo k chybnému zjištění, že se dva objekty protínají, i když tomu tak není, a proto nelze použít tuto metodu pro nekonvexní objekty.

2.2.2 BVH

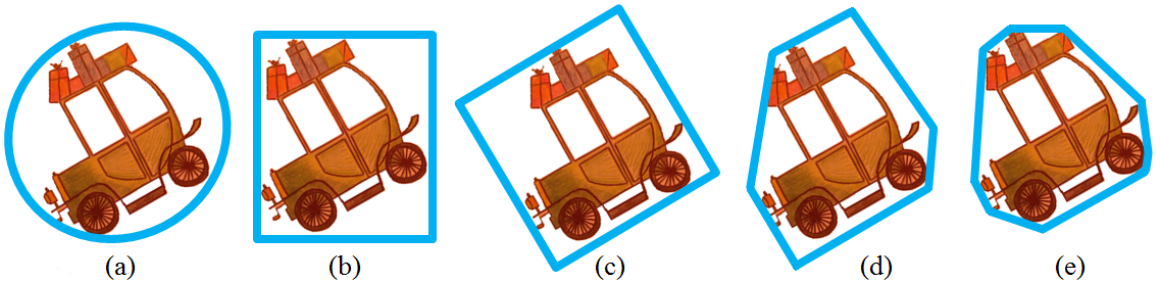
Zkratka BVH (z angl. boundary volume hierarchy) představuje hierarchii obálek. Ericson a Bergen [16, 33] popisují princip pomocné datové hierarchie obálek, která je graficky znázorněna pomocí stromu a která slouží pro nahrazení geometrie původního objektu zvoleným typem obálky pro urychlení různých druhů výpočtů (např. detekce kolizí). Na obrázku 2.10 je znázorněna scéna, která je tvořena dvěma objekty, jejichž povrch je reprezentován trojúhelníkovou sítí. Pro vyobrazenou scénu jsou znázorněny BVH stromy pro oba dva objekty. Reference na základní grafická primitiva, které tvoří povrch objektu, jsou uložena v listech stromu. Pro stavbu stromu odspoda nahoru se jedná o proces shlukování obálek. Nejprve se určí obálky OBB_0 nad grafickými primitivy a poté se určí obálky nad obálkami OBB_0.



Obrázek 2.9: (vlevo) Správný výsledek SAT. (vpravo) Nesprávný výsledek SAT. Zelená přímka je separační přímkou, růžová přímka je separační osou a žlutá úsečka představuje vytvořený interval.



Obrázek 2.10: Na obrázku pro případ (a) je znázorněn příklad BVH pro dvojici 3D objektů. Reference na grafická primitiva (např. trojúhelníky) jsou obsaženy v listech stromu. Pro případ (b) je zobrazen kořen stromu, který je označen písmenem A, reprezentující obálku, která aproximuje první objekt. Pro případ (c) je zobrazen kořen stromu, který je označen písmenem F, reprezentující obálku, která obsahuje druhý objekt.



Obrázek 2.11: Příklady typů obálek ve 2D používaných pro techniku BVH: (a) kruhová obálka, (b) osově zarovnaná obálka (AABB), (c) orientované obálky (OBB), (d) k–DOP, kde k=6, (e) konvexní obálka.

Dochází tedy k propagaci obálek, až dojde ke stanovení obálky kořene stromu. Pro případ konstrukce stromu odshora dolů se postupuje opačně. Nejprve se vytvoří obálka kořene stromu, který zahrnuje všechny obálky ve scéně. Poté se určí dle definované metody (např. prostorový medián, objektový medián) dle jaké souřadnicové osy dojde k rozdělení obálky a vytvoří se levý a pravý podstrom. Dělení prostoru obálky do menších obálek, které leží uvnitř obálky rodiče, končí v případě, kdy jsou splněny ukončovací pravidla pro stavbu BVH. Ukončovacími pravidly může být například maximální povolený počet trojúhelníků v listu BVH a/nebo maximální hloubka stromu. Pro každý 3D objekt ve scéně se vytvoří obálka. Ericson a Bergen [16, 33] prezentují několik typů obálek (viz obrázek 2.11), které jsou používány pro techniku BVH, a jejichž vlastnosti a princip tvorby bude popsán v následujícím textu. Detekce kolizí za použití hierarchie obálek je konzervativním testem. Jedná se o konzervativní test z důvodu, že pokud se obálky neprotnou, pak neexistuje ani jeden společný průsečík objektů. Pokud se však obálky protnou, pak možná existuje průsečík objektů. Dojde k průchodu stromu a zjištění, zda dochází k protnutí obálek pro jednotlivé 3D objekty. Pokud dojde k protnutí obálek, pak je potřeba projít základní grafická primitiva obsažená v této obálce, většinou jsou jimi trojúhelníky, a zjistit, zda se tyto trojúhelníky dvojice testovaných objektů protínají, aby bylo možné stanovit, zda dochází ke kolizi objektů. Pojďme se nyní podívat na přehled typů obálek, které prezentují Ericson a Bergen [16, 33].

Kruhová/kulová obálka (viz obrázek 2.11(a)) je reprezentována pomocí souřadnic středu koule v prostor a jejího poloměru. Pro tento typ obálky je rychlý test, zda se dvě kulové obálky protínají, který se provede tak, že se spočítá vzdálenost středů obou koulí a pokud je tato vzdálenost větší nežli součet kvadrátů poloměrů obou koulí, pak se koule neprotínají. V opačném případě dochází k průniku testovaných koulí. Tato obálka však těsně neobepíná kontury objektu. Tato vlastnost se s výhodou využívá pro animaci ve scénách, kde při provedení transformace rotace objektu okolo vlastní osy rotace, pokud se jedná o rotační tělesa, se obálka součásti nemění. Při aplikování transformace posunu objektu dojde k pouhému posunu středu koule, tzn. je jednoduché aktualizovat tuto obálku. Nevýhodou této obálky je, že díky skutečnosti, že obálka neobepíná těsně objekt a objekt by nekonal pohyb, je povětšinou volný prostor uvnitř koule příliš velký, tzn. test průniku dvou koulí není moc reflektující pravděpodobnost průniku dvou objektů.

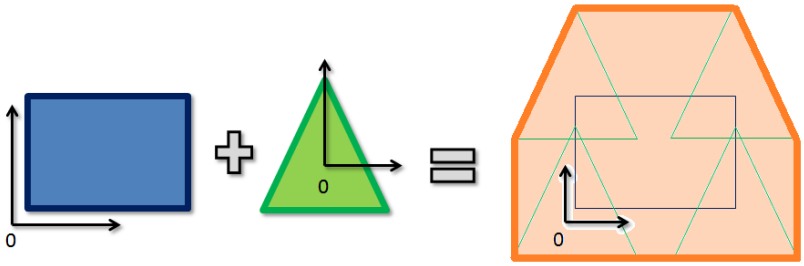
AABB (z angl. axis aligned bounding box) obálka (viz obrázek 2.11(b)) neboli osově zarovnaná obálka má zarovnány stěny s osami globálního souřadnicového systému. Test průniku dvou AABB obálek se provede na základě určení, zda se protínají intervaly těchto dvou obálek. Pro 3D existují pro každý AABB tři intervaly ve směru souřadnicových os. Pokud se AABB obálky neprotínají, pak nedochází k tomu, že by existoval průnik intervalů AABB pro všechny tři souřadnicové osy. Výhodou této obálky je, že její konstrukce je rychlá a že lépe obepíná 3D objekt oproti kouli.

Dalším typem obálky je orientovaná obálka (z angl. oriented bounding box), která je také často označována zkratkou OBB. Stěny OBB obálky (viz obrázek 2.11(c)) jsou libovolně orientované. Tento druh obálky dobře obepíná aproximovaný objekt. Stanovení souřadnic takovéto obálky probíhá na základě určeného středu této obálky, který se určí z vypočtené konvexní obálky vrcholů objektů. Vypočte se kovarianční matice, vlastní vektory a vlastní čísla kovarianční matice. Určí se rozměry tělesa ve směru vlastních vektorů, které poslouží pro stanovení délky hran OBB. Souřadnice vrcholů OBB se vypočtou ze znalosti těžiště OBB a přičtením/odečtením poloviční délky hrany ve směru příslušného vlastního vektoru. Pro určení, zda se dvě OBB obálky protínají, slouží technika, která se nazývá SAT (viz podkapitola 2.2.1).

K–DOP (z angl. discrete orientation polytope) obálka (viz obrázek 2.11(d)) je zadána pomocí k/2 pevně stanovených směrů, které jsou normalizovanými vektory (d_1, d_2,$\cdots$,$d_{k/2}$). Polytope je zadán pomocí normál stěn (d_1, $\cdots$, $d_{k/2}$, -d_1, $\cdots$, -$d_{k/2}$) a je reprezentován $k/2$ intervaly. Test průsečíků dvou k–DOPů probíhá na základě určení průsečíků intervalů, které tvoří k–DOP. Pokud neexistuje žádný průsečík intervalů k–DOPů, pak se objekty neprotínají. Pokud však existuje průsečík alespoň jednoho intervalu k–DOPů, pak je možné, že se objekty mohou protínat.

Konvexní obálky (viz obrázek 2.11(e)) nejoptimálněji obepínají aproximovaný objekt. Test průniků dvou konvexních obálek je pomalý, jelikož nemá konvexní obálka jednoduchou reprezentaci. Povětšinou jde bohužel ruku v ruce těsnější aproximace objektu s náročnějším testem, zda se dvě konvexní obálky protínají. Je nutné nalézt takový kompromis pro určení typu obálky, který bude přijatelný pro konkrétní použití a který bude vyhovovat konkrétním požadavkům na příslušnou aplikaci.

Chang a kol. [12] prezentují princip metody pro detekci kolizí, která je založena na vytvoření dvou typů obálek (koule a OBB) a která tak kombinuje výhody obou dvou použitých obálek. Vytvoří se strom OBB-kulového hraničního objemu tak, že ve všech vnitřních uzlech bude vytvořena jak OBB tak kulová obálka. Rychlost detekce kolizí závisí na tom, jak dobře aproximuje obálka objekt a jak je rychlý test existence průsečíku dvou testovaných obálek. Nejprve je použit test detekce kolizí pro dvě kulové obálky, protože test detekce kolizí tohoto typu obálky je velmi rychlý. Pokud neexistuje alespoň jeden průsečík dvou koulí, pak se objekty neprotínají. V opačném případě bude pro objekty proveden test detekce kolizí jejich OBB obálek použitím SAT teorému, jehož princip byl vysvětlen v podkapitole 2.2.1. Pokud bude nalezen průsečík dvou OBB obálek, pak bude proveden přesný test detekce kolizí nad množinami trojúhelníků testovaných objektů.



Obrázek 2.12: Minkowského součet: Jako objekt pro zvětšení původního objektu (obdélníku) je využit znázorněný rovnoramenný trojúhelník. U každého objektu je znázorněn lokální souřadnicový systém, ve kterém je definován objekt.

2.2.3 Minkowského součet

Halperin [17] prezentuje metodu, která stanoví, zda existuje kolize analyzované dvojice 3D objektů (O_1, O_2), na základě zjištění, zda existuje kolize objektu O_1 zvětšeného pomocí principu Minkowského součtu o objekt O_2.

Formální zápis Minkowského součtu, kterou Halperin [17] používá pro stanovení, zda existuje kolize dvojice testovaných objektů, je

$$A + B = \{x + y : x \in A, y \in B\}, \tag{2.4}$$

kde A představuje množinu bodů, které obsahuje jeden objekt, a B představuje množinu bodů, které obsahuje druhý testovaný objekt. Množina bodů obsahuje body uvnitř a na povrchu objektu.

Po vypočtení zvětšeného tvaru objektu O_{1V} stačí provést test, zda bod objektu O_2 ležící v jeho počátku lokálního souřadnicového systému protne objekt O_{1V}. Pokud existuje jejich průsečík, pak existuje kolize testované dvojice objektů O_1, O_2, jinak kolize neexistuje. Pokud jsou oba dva objekty konvexní, pak i výsledek dilatace (zvětšení) je konvexní objekt. Princip Minkowského sumy je uveden na obrázku 2.12, ve kterém je znázorněno, že počátek lokálního souřadnicového systému rovnoramenného trojúhelníku je umísťován do bodů z množiny bodů, které obsahuje obdélník, a je vypočten zvětšený obdélník o rovnoramenný trojúhelník.

2.2.4 Detekce kolizí založena na voxelizaci

Pro rychlé stanovení, zda leží vrchol uvnitř 3D objektu, je vhodné mít přímé informace o objemu testovaného 3D objektu. Pokud je objekt reprezentován pomocí povrchové reprezentace (např. trojúhelníkovou sítí), pak je známa pouze struktura jeho povrchu a neexistuje pro tento objekt přímo informace o jeho vnitřku. Proces získání objemové reprezentace objektu z jeho známé povrchové reprezentace se nazývá voxelizace [36]. Objemová reprezentace objektu je tvořena objemovými daty. Objemová data jsou skalárními daty a mohou být strukturována do více podob, např. do pravidelné nebo nepravidelné mřížky. Skalární data mohou představovat například informaci o hustotě 3D objektu pro data získaná z počítačové

tomografie (CT). Objemový element (voxel) představuje krychli, která je nejmenším elementem ve vytvořeném trojrozměrném diskrétním prostoru. Objemová data obsahující informaci o existenci objemu objektu v určitém místě jeho objemu a budou uložena v pravidelné pravoúhlé prostorové mřížce. Problematika detekce kolizí objektů má za úkol stanovit, zda dochází ke kolizi (střetnutí) testovaných objektů. Pokud existuje kolize analyzované dvojice objektů, pak si lze tuto skutečnost představit tak, že se objekty vzájemně protínají, což znamená, že dochází ke kolizi jednoho či více objemových elementů dvojice objektů, ve kterých existuje objem obou dvou uvažovaných objektů. Objemová data mají větší paměťové nároky, lze nad nimi snadno provádět logické operace a lze snadno pracovat s informací o objemu objektu.

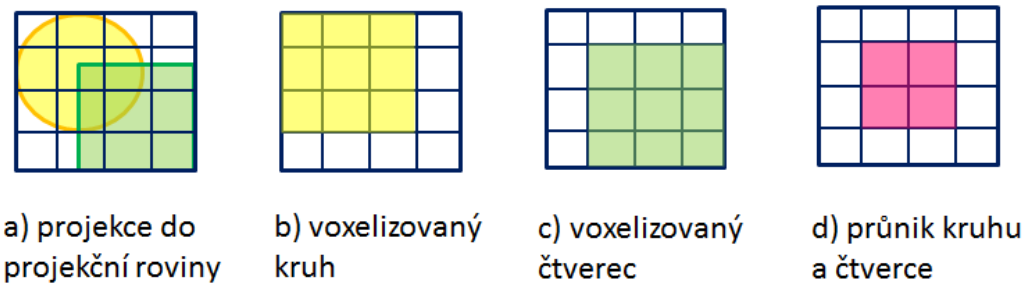
Eisemann a kol. [15] prezentují metodu detekce kolizí na grafickém procesoru, která je založena na voxelizaci 3D objektů. Objemová reprezentace objektu je uložena do 2D textury a je získána jeho voxelizací pomocí bitových operátorů. Prvek textury se jmenuje texel (z angl. texture element). Každý texel je reprezentován jako RGBA hodnota. Vizuálně si lze představit texel jako barevný pixel (RGB kanály), který může být průhledný (A kanál). Voxel představuje analogii k dvourozměrnému pixelu. 2D textura představuje binární mřížku voxelů. Jednotlivé bity barevných kanálů jsou chápány jako voxely a jsou použity pro uložení informace, zda existuje objem 3D objektu v daném místě mřížky. 2D textura, jejíž prvky jsou typu RGBA, představuje krychli, jejíž přední stěna leží v rovině xy a jejíž prvky představují řezy (z angl. slice) ve směru souřadnicové osy z (viz obrázek 2.16). Popis principu voxelizace je uveden v podkapitole 2.2.5.

V následujícím textu budou popsány principy dvou metod pro stanovení detekce kolizí, které jsou založené na znalosti objemové reprezentace testovaných objektů. První dále uvedená metoda, která je založena na detekci kolizí za použití voxelizace obou objektů, stanoví, zda existuje či neexistuje kolize objemových dat uvažovaných objektů. Druhá prezentovaná metoda pro detekci kolizí dvojice objektů je založena na testování existence kolize vrcholů objektu s voxelizovaným objektem. V této práci je realizována druhá uvedená technika z důvodu požadovaného budoucího rozšíření této práce pro stanovení, kde se objekty protínají.

Detekce kolizí za použití voxelizace obou objektů

Eisemann a kol. [15] prezentují princip metody pro zjištění, zda existuje kolize objemových dat dvojice objektů. Princip této metody je znázorněn na obrázku 2.13. Nejprve se získá objemová reprezentace obou testovaných objektů, která bude uložena do dvou oddělených 2D textur, které představují binární pole voxelů (viz obrázek 2.13(b,c)). Princip získání objemové reprezentace objektu ze znalosti jeho povrchové reprezentace je vysvětlen v podkapitole 2.2.5. Pro stanovení, zda existuje kolize objemových dat uvažované dvojice objektů, je potřeba smísit objemové informace obou testovaných objektů do jedné výsledné 2D textury, ve které pokud budou existovat nenulové bity některého texelu, pak existuje kolize analyzované dvojice objektů. Aby existoval nenulový bit ve výsledné textuře pouze v texelu, na jehož pozici kolidují objemová data testovaných objektů, je potřeba smísit objemová data obou objektů použitím bitového operandu bitového součinu. Výsledná 2D textura obsahuje nenulové hodnoty pouze takových bitů texelů, které kolidují pro analyzovanou dvojici objektů (viz obrázek 2.13(d)).

Všechny čtyři 2D textury musí mít stejné rozlišení (viz obrázek 2.13 rozlišení 4^3), aby je bylo možné smísit a také proto, aby výsledek detekce kolizí uvažované dvojice objektů byl



Obrázek 2.13: Princip detekce kolizí za použití voxelizace zkoumané dvojice 3D objektů. Objemová data objektů (b,c). Výsledná textura obsahuje logické jedničky pouze na takových bitech, kde objekty kolidují (d).

správný. Protože není žádoucí, aby výsledek voxelizace objektu byl závislý na pozici kamery, musí být použita ortogonální (paralelní) projekce, která představuje ve 3D prostoru krychli. Protože je pro detekci kolizí potřeba zachovat prostorové uspořádání analyzované dvojice objektů, musí být provedena voxelizace obou objektů pro stejnou oblast 3D prostoru. Tuto skutečnost si lze představit tak, že pro analyzovanou dvojici objektů se vytvoří OBB obálka, která má tvar krychle a která bude obklopovat oba analyzované objekty (viz obrázek 2.13(a) modrý čtverec). Oba objekty budou voxelizovány uvnitř tohoto vymezeného prostoru.

Pomocí této metody není možné určit souřadnice průsečíku dvou objektů, ale je možné pouze stanovit, zda existuje či neexistuje kolize pro analyzovanou dvojici 3D objektů.

Detekce kolizí založená na testování existence kolize vrcholů objektu s voxelizovaným objektem

Nechť O_1 je voxelizovaný objekt. Označme modelovací a pohledovou matici použitou pro voxelizaci objektu O_1 jako $(MV)_1$ a jeho projekční matici jako P_1. Pro transformaci vrcholů testovaného druhého objektu jsou použity stejné matice $((MV)_1, P_1)$, které byly použity pro voxelizaci prvního objektu. Tento postup je nutné provést z důvodu, že je potřeba zachovat výchozí prostorové uspořádání 3D objektů.

V této práci je realizována technika detekce kolizí dvojice analyzovaných objektů na grafickém procesoru, která je založena na principu již popsané metody detekce kolizí za použití voxelizace obou objektů [15]. Z důvodu požadavku na znalost souřadnic průsečíků dvojice objektů pro budoucí rozšíření této práce, byla upravena popsaná metoda detekce kolizí za použití voxelizace obou objektů [15] tak, že metoda detekce kolizí dvojice objektu P_1, P_2 je tvořena dvěma průchody, kde v každém průchodu bude nejprve voxelizován objekt P_v a poté bude otestováno, zda alespoň jeden vrchol z množiny vrcholů $\mathcal{V}$ součásti P_t koliduje s objemovými daty objektu P_v, kde v,$t \in \{1, 2\}$ a $\mathcal{V}$ představuje buď množinu všech vstupních vrcholů $\mathcal{A}$ součásti P_t a nebo množinu extrémních vrcholů $\mathcal{E}$ konvexní obálky součásti P_t.

Nejprve je otestována existence kolize vrcholů množiny $\mathcal{E}$ součásti P_t s objemovými daty součásti P_v. Pokud neexistuje jejich kolize, tak bude otestováno, zda existuje kolize vrcholů množiny $\mathcal{A}$ součásti P_t s voxelizovanou součástí P_v. Tento přístup byl zvolen z důvodu, že

mohutnost množiny $\mathcal{E}$ je markantně menší nežli je mohutnost množiny $\mathcal{A}$. Případné otestování existence kolizí vrcholů součásti P_t, které nejsou extrémními vrcholy její konvexní obálky, s objemovými daty součásti P_v, bylo navrženo z důvodu, že test detekce kolizí extrémních vrcholů součásti P_t s objemovými daty součásti P_v není vždy postačující (viz obrázek 2.14). Na obrázku 2.14(uprostřed) jsou zobrazeny fialovou barvou extrémní vrcholy konvexní obálky součásti. Vrcholy, které leží na povrchu objektu, jsou znázorněny na obrázku 2.14(vpravo) červenou barvou. Pokud bude nalezena kolize testované dvojice objektů, pak již nedojde k opětovnému testování existence detekce kolize příslušné dvojice součástí, tzn. dojde k ukončení provádění dále popsaných testů pro danou dvojici součástí.

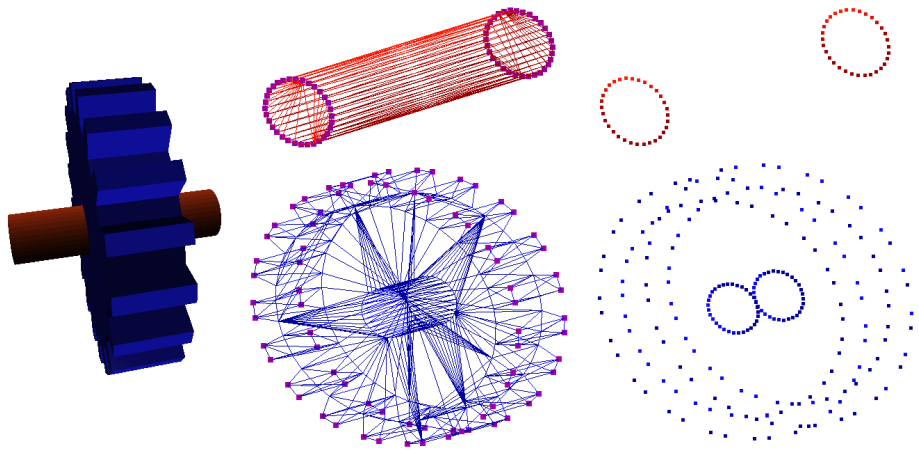
V prvním průchodu algoritmu bude provedena detekce kolizí s následujícími hodnotami: P_v=P_1, P_t=P_2 a $\mathcal{V}$=$\mathcal{E}$. Pokud nebyla detekována kolize analyzované dvojice součástí po provedení detekce kolizí s uvedenými parametry, tak se otestují místo extrémních vrcholů součásti P_2 vrcholy, které leží na jejím povrchu a které jsou označeny jako množina $\mathcal{A}$. V druhém průchodu bude voxelizován objekt, jehož vrcholy byly v prvním průchodu testovány (P_v=P_2), budou testovány vrcholy objektu, který byl v prvním průchodu voxelizován (P_t=P_1), a množina testovaných vrcholů je stejná jako v prvním průchodu. Pokud po provedení obou průchodů pro detekci kolizí nebyla nalezena kolize analyzované dvojice součástí, pak neexistuje jejich kontakt.

Pro účel detekce kolizí jsou extrudovány jak vrcholy testovaného objektu P_t tak i vrcholy voxelizovaného objektu P_v z důvodu, že má být detekován kontakt i pro součásti, které jsou od sebe vzdálené o stanovenou velikost. Extrudování vrcholu se provede ve směru jeho výsledného normálového vektoru, který je stanoven jako aritmetický průměr normál stěn, na níž leží analyzovaný vrchol, o definovanou konstantu. Nelze extrudovat vrcholy pouze ve směru normál stěn, protože pokud bude vrchol náležet alespoň dvěma stěnám, které budou mít odlišné normálové vektory, pak by extrudováním vrcholu ve směru normály stěny došlo k rozštěpení geometrie objektu, protože by extrudováním jednoho vrcholu vzniklo právě tolik nových vrcholů, kolika stěnám by daný vrchol náležel. Pro zajištění správné detekce kolizí součástí soustavy je nezbytné, aby normály součásti byly správně orientovány, tzn. aby směřovaly směrem vně objektu.

Detekce kolizí je testována pro každou dvojici objektů ve scéně. Celkový počet testů objektů je roven kombinaci bez opakování dvojic z n prvků. Kombinace bez opakování k-tic z n prvků se vypočítá dle následujícího vzorce

$$C_k(n) = \binom{n}{k} = \frac{n!}{k!\,(n-k)!}. \tag{2.5}$$

Stanovení, zda vrchol součásti koliduje s voxelizovanou součástí, se provádí na grafickém procesoru. Vertex shader bude zavolán na každý vrchol součásti, který bude testován pro detekci kolizí s voxelizovanou součástí. Aby bylo možné určit, jaký vrchol součásti koliduje s voxelizovaným objektem, je potřeba při výpočtu, zda existuje kolize testované dvojice součástí, znát unikátní identifikační číslo aktuálně testovaného vrcholu součásti. Pozice ve směru osy–z testovaného vrcholu bude zakódována do 1D textury, jejíž elementy budou typu RGBA. Tato textura bude obsahovat logickou jedničku pouze na pozici bitu, který představuje řez v binární 3D mřížce, kam spadá testovaný vrchol. Elementy textury představují graficky řezy binární 3D mřížky, ve které je uložena informace o objemu voxelizovaného 3D objektu. Detekce kolize dvojice součástí se provádí použitím bitového operátoru bitového



Obrázek 2.14: Úspěšnost detekce kolizí dvojice součástí (vlevo). (uprostřed) Neúspěšný test: extrémní vrcholy konvexní obálky objektu P_t a voxelizovaný objekt. (vpravo) Úspěšný test: vrcholy objektu P_t a voxelizovaný objekt.

součinu. Bitový součin bude aplikován na 1D texturu, která obsahuje zakódovanou pozici testovaného vrcholu, a na texel binární 3D mřížky, jehož hodnota v rovině xy odpovídá souřadnicím x, y testovaného vrcholu. Informace o kolidujících testovaných vrcholech se uloží do výsledné 1D textury, kde do každého jejího bitu je zakódována informace, zda testovaný vrchol s příslušným indexem koliduje s voxelizovaným objektem či nikoliv. Pokud po provedení testu detekce kolizí existuje alespoň jeden kolidující vrchol testovaného objektu s voxelizovaným objektem, pak analyzovaná dvojice součástí má kontakt.

Omezení realizované metody, která je založená na testování existence kolize vrcholů objektu s voxelizovaným objektem

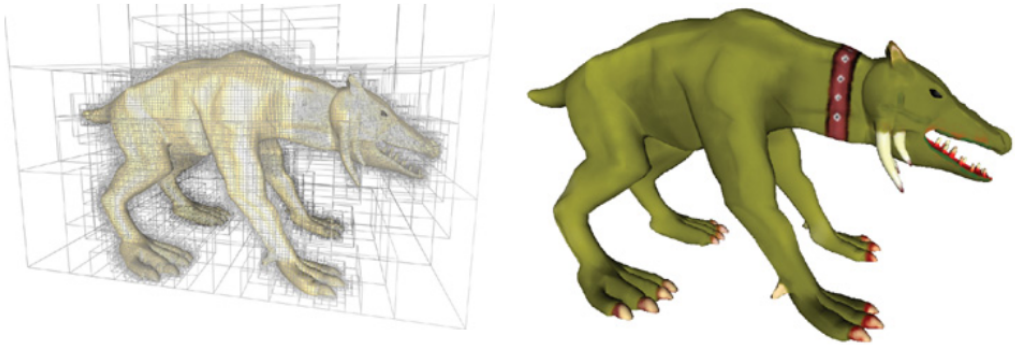
Tato metoda vyžaduje, aby grafický procesor podporoval bitové operátory a celočíselné textury. Aby byla získána korektní informace o párech součástí soustavy, které spolu mají kontakt, je nutné, aby byly normály 3D modelů správně orientovány, tzn. aby směřovaly vně objektu.

2.2.5 Voxelizace

Voxelizací objektu rozumíme získání objemové reprezentace objektu ze znalosti jeho povrchové reprezentace. V následujícím textu bude nejprve uveden princip metody voxelizace objektu s použitím techniky vrhání paprsků. Poté bude uvedena metoda voxelizace objektu s použitím bitových operátorů, kterou prezentují Eisemann a kol. [15] a která je realizována v této práci.

Voxelizace objektu s použitím techniky vrhání paprsků

Nooruddin a kol. [28] prezentují metodu, která vytvoří z povrchové reprezentace modelu její objemovou reprezentaci použitím metody vrhání paprsků. Paprsek je jednoznačně ur-



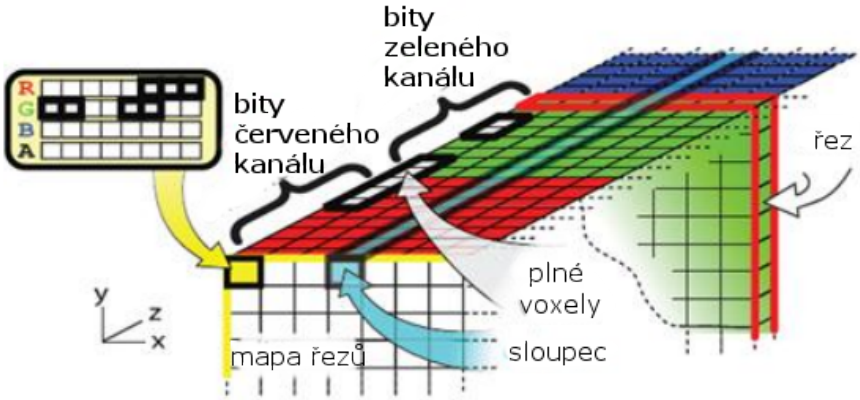
Obrázek 2.15: (vlevo) Objemová reprezentace 3D objektu, který je znázorněn na obrázku vpravo. Zdroj obrázku [14].

čen svým působištěm, směrovým vektorem a velikostí (pokud se neuvažuje normalizovaná velikost). Nejprve by došlo ke stanovení osově zarovnané obálky (AABB), která by obalovala voxelizovaný model. Výsledný voxelizovaný objekt by byl uložen například do datové struktury oktantového stromu. Oktantový strom je takový strom, jehož každý vnitřní uzel obsahuje vždy osm potomků. Vnitřní uzel je takový objemový element, který není celý uvnitř či vně analyzovaného objektu. Listem v oktantovém stromu je takový objemový element, který je celý uvnitř či vně analyzovaného objektu. Bylo by stanoveno rozlišení voxelizace, které udává ukončující podmínku pro dělení vytvořeného vnitřního uzlu oktantového stromu. Došlo by k rozdělení AABB obálky vždy v polovině intervalu každého směru jeho souřadnicové osy. Pro 3D případ by tímto rozdělením vzniklo vždy osm nových objemových elementů. Vzniklé objemové elementy v osově zarovnaném kvádru nazveme voxely. Pro každý voxel by se vygenerovalo několik uniformě rozložených působišť paprsku uvnitř voxelu a pro každé působiště by se vygenerovalo náhodně několik směrových vektorů paprsku. Poté by došlo k vrhnutí vytvořených paprsků a otestování, kolik trojúhelníků a v jakých místech protíná paprsek objekt. Stanovilo by se, zda je větší množina, která má sudý počet průsečíků či lichý počet průsečíků. Pokud bude větší množina se sudým počtem průsečíků, pak leží voxel vně objektu. V opačném případě leží voxel uvnitř objektu. Jak vypadá výsledek voxelizace objektu za pomocí takto vytvořeného voxelizovaného modelu, vidíme na obrázku 2.15.

Voxelizace objektu s použitím bitových operátorů

V této práci je realizována technika voxelizace na grafickém procesoru, kterou prezentují Eisemann a kol. [15] a jejíž princip bude nyní vysvětlen. Pro voxelizaci na GPU se používají bitové operátory bitová nonekvivalence a bitový součet. Pomocí bitové nonekvivalence se získá vnitřní objem analyzovaného objektu a pomocí bitového součtu se získají kontury analyzovaného objektu. V této práci je realizována voxelizace objektu pouze s využitím bitového operátoru bitové nonekvivalence z důvodu, že dochází k extrudování objektu pro účel detekce kolizí, tzn. objemová reprezentace extrudovaného objektu bude obsahovat kontury objektu.

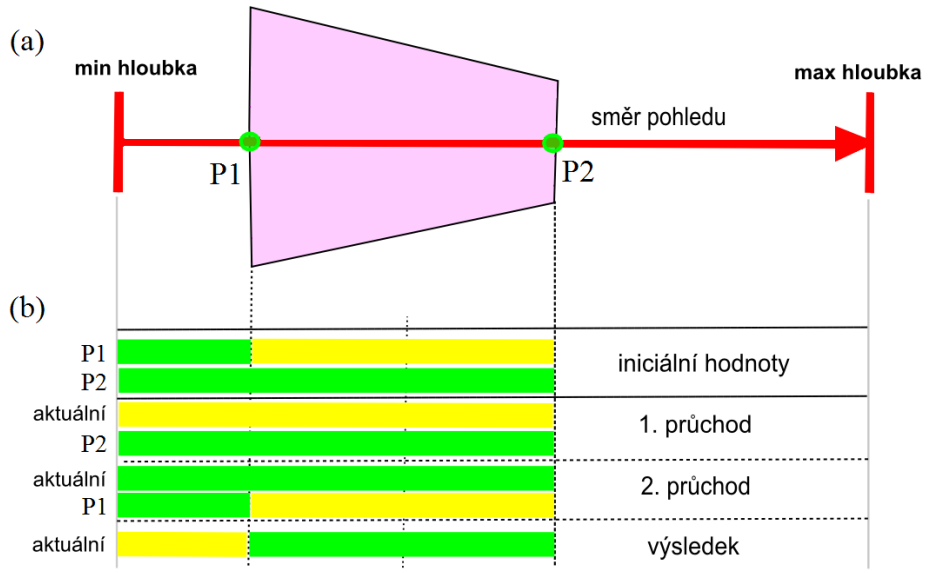
Objemová data voxelizovaného objektu se uloží do binární 3D mřížky voxelů, které jsou



Obrázek 2.16: Mapa řezů je binární 3D mřížka, která je reprezentována 2D texturou. Každý texel obsahuje RGBA hodnotu, jehož bity slouží pro uložení binární informace, která představuje informaci o tom, zda v dané části 3D prostoru existuje objem voxelizovaného objektu. Texty obrázku byly přeloženy do češtiny autorkou práce. Zdroj obrázku [15].

reprezentovány jako 2D textura. Popisovaná situace je znázorněna na obrázku 2.16. Řekněme, že index i reprezentuje index řádku voxelu, index j reprezentuje index sloupce voxelu a index k představuje index řezu voxelu. Voxel se souřadnicemi (i,j,k) může nabývat hodnoty binární 0 (voxel leží vně objektu) nebo 1 (voxel leží uvnitř objektu) a je zakódován do k–tého bitu RGBA hodnoty reprezentující pixel (i,j). Pro získání binární 3D mřížky s objemovými daty objektu je potřeba vykreslit objekt s použitím bitové nonekvivalence do framebufferu, který je spojen s 2D texturou. Hodnota bitu ve výsledném binárním čísle na příslušné pozici je po aplikování bitové nonekvivalence rovna logické jedničce, pokud jsou odlišné hodnoty bitů na stejné pozici ve vstupních binárních číslech. Nechť n fragmentů, které tvoří povrch objektu a které mají stejné souřadnice i a j, leží před voxelem (i,j,k), pak bude voxel ležet uvnitř objektu, pokud je číslo n lichých číslem. Představme si, že každý voxel obsahuje inkrementální čítač a každý fragment zvětší hodnotu čítače o jedna pro všechny voxely, které leží před ním. Místo toho, aby byla na výstupu shaderu hodnota, která bude mít pouze jeden bit jedničkový na k–té pozici na základě hloubky zpracovávaného fragmentu, budou získány jedničkové bity pro všechny pozice menší nežli k. Přidáním tohoto sloupce do příslušného sloupce čítače ve voxelové mřížce, inkrementujeme hodnoty ve všech pozicích takových voxelů, jimiž prochází paprsek ze směru pozorovatele pro zpracovávaný fragment. Bude použita vstupní maska založená na zpracovávaném fragmentu, která je kumulována pomocí bitové operace bitové nonekvivalence. Bitová maska je realizována ve fragment shaderu jako vyhledávací tabulka v 1D textuře založené na hloubce fragmentu.

Příklad aplikování popsaného principu voxelizace pro 3D objekt tvaru komolého kužele, jehož řez v rovině, ve které leží směrový vektor znázorněného paprsku, je uveden na obrázku 2.17. Paprsek, který je znázorněn na obrázku červenou šipkou, protíná kontury objektu ve dvou bodech, které jsou označeny jako *P1* a *P2*. Na obrázku 2.17(b) je zobrazen obsah framebufferu pro pixel, kterým prochází zobrazovaný paprsek a který je označen jako *aktuální*. Zelená barva představuje hodnotu logické 1 (voxel leží uvnitř objektu) a žlutá barva před-

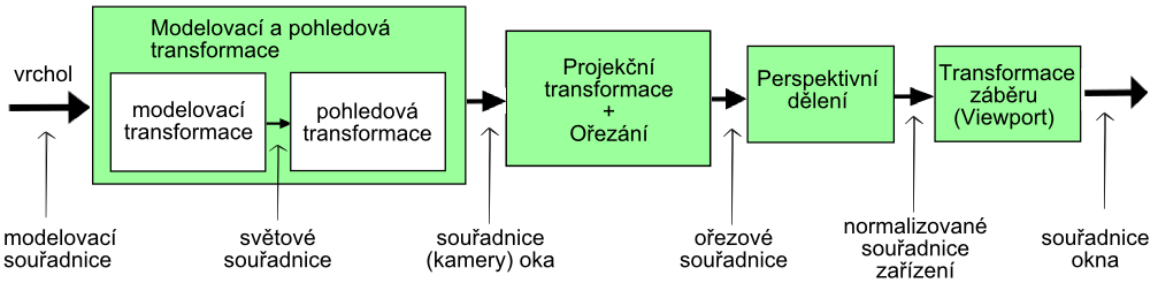


Obrázek 2.17: Voxelizace s použitím bitové nonekvivalence: Žlutá/zelená barva – vně/uvnitř objektu. Hodnoty pomocné textury jsou označeny jako iniciální hodnoty a představují nalezené průsečíky povrchu objektu s paprskem. Bitová nonekvivalence se aplikuje na iniciální hodnotu a aktuální hodnotu ve framebufferu.

Obrázek 2.18: Příklad zjednodušeného zobrazovacího řetězce.

stavuje hodnotu logické 0 (voxel leží vně objektu). Aby se fragmentový procesor zavolal na oba dva nalezené průsečíky (i zakryté průsečíky objektem), je potřeba vypnout hloubkový test. Pořadí zpracování fragmentů není přesně určeno z důvodu možných vzniků zpoždění a hazardů v zobrazovací pipeline OpenGL. Nejprve se nalezne, kolik fragmentů leží před průsečíky *P1*, *P2*. Tato informace je označena v obrázku jako *iniciální hodnoty*. V prvním průchodu je hodnota *aktuální* rovna iniciální hodnotě (všechny bity nulové). Bitová nonekvivalence je aplikována v každém průchodu na hodnotu označenou jako *akutální* a aktuálně zpracovaný průsečík. Po zpracování všech průsečíků jsou v proměnné označené jako *aktuální* obsažena objemová data 3D objektu v daném pixelu, jímž prochází vyslaný paprsek.

Pro účel voxelizace je potřeba umístit voxelizovaný objekt do takové požadované polohy, aby výsledek voxelizace nebyl závislý na pozici, natočení a velikosti načteného 3D objektu ve scéně. Protože v této práci jsou naprogramovány některé části programovatelného řetězce GPU (vertex a fragment shader), je potřeba vědět, jaké transformace jsou aplikovány na každý vrchol vykreslovaného 3D objektu. Popis zobrazovacího řetězce OpenGL a aplikované



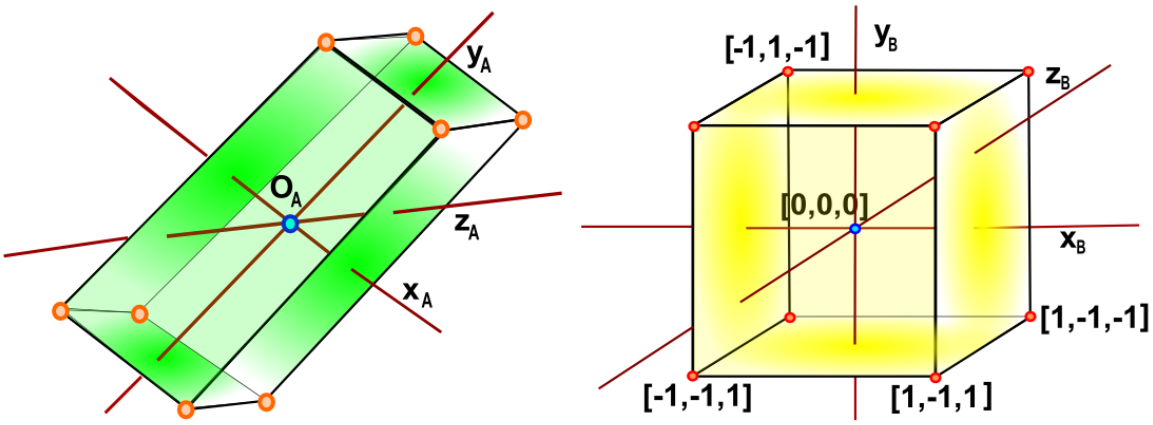
Obrázek 2.19: Kroky při transformaci vrcholů renderovaného objektu.

transformace pro zarovnání objektu do požadované polohy pro jeho voxelizaci je uveden v následujícím textu.

Zobrazovací řetězec OpenGL

Shader je zdrojový kód, který slouží k řízení jednotlivých částí programovatelného vykreslovacího řetězce grafické karty (viz obrázek 2.18). Programovací jazyk, který byl použit v této práci pro tvorbu shaderů, je jazyk GLSL pro použitou knihovnu OpenGL. OpenGL knihovna přestavuje svým chováním stavový automat. Vykreslovací řetězec definuje posloupnost příkazů, které z dat definující scénu vytvoří dvourozměrný obraz dané scény. Jednotlivé kroky (viz obrázek 2.18) zobrazovacího řetězce nazveme stupni grafického řetězce. Nejprve jsou na každý vrchol vykreslovaného objektu aplikovány transformace uvedené na obrázku 2.19. Poté jsou na základě znalosti topologie součásti vytvořeny trojúhelníky pro transformované vrcholy ve 2D, které tvoří stěny vykreslovaného 3D objektu. Při rasterizaci trojúhelníku dojde k jeho případnému oříznutí či odstranění jeho částí, které jsou umístěny mimo výřez obrazovky, a zbylé viditelné části trojúhelníku jsou rasterizovány do fragmentů, které mají velikost pixelu. Výstupem fragment shaderu je barva a hloubka zpracovaného fragmentu, který bude vykreslen do framebufferu, pokud projde všemi povolenými testy (např. hloubkový test). Grafický procesor, který je součástí grafické karty, vykoná posloupnost příkazů, které vytvoří dvourozměrný obraz vykreslované scény. Konkrétní podoba vykreslovacího řetězce se neustále vyvíjí s ohledem na vývoj trendů v oblasti počítačové grafiky. Dalo by se říci, že nejzásadnější změnou ve vývoji vykreslovacího řetězce je přechod od fixního vykreslovacího řetězce k programovatelnému řetězci. Tento přechod byl způsoben příchodem třetí generace grafických karet (např. GeForce 3 od firmy NVIDIA). Ve fixním vykreslovacím řetězci bylo pevně dáno chování jeho jednotlivých stupňů. Pro programovatelné řetězce je možnost stanovit podle konkrétní potřeby pro vyvíjenou aplikaci chování jednotlivých stupňů řetězce pomocí shaderů. V OpenGL se objekty definují svými vrcholy a objekt se transformuje po vrcholech. Transformace se vypočítají jako maticové násobení souřadnic vrcholu a příslušné transformační matice. Transformace vrcholů v OpenGL se skládá z následujících kroků [34], které jsou zobrazeny na obrázku 2.19.

1. Modelovací transformace určí umístění a natočení objektu ve scéně.



Obrázek 2.20: (vlevo) Objekt A, který se bude voxelizovat. (vpravo) Ořezový prostor, do kterého se transformuje objekt A.

2. Zobrazovací (pohledová) transformace stanoví polohu, natočení a směr pohledu kamery (pozorovatele).
3. Projekční transformace nastaví tvar pohledového objemu (kvádr pro paralelní projekci či komolý kužel pro perspektivní projekci) a ořezové souřadnice. Pohledový objem určí, jak se objekt promítá na průmětnu. Ořezové souřadnice stanoví polohu ořezávacích rovin, které určují, jaké objekty či jejich části budou vykresleny.
4. Perspektivní dělení se uplatní pouze pro perspektivní projekci, protože pro paralelní projekci je hodnota perspektivního dělení rovna 1.
5. Transformace pracoviště (viewport) stanoví obdélníkovou oblast okna, do kterého se namapuje průmětna.

Zarovnání objektu pro voxelizaci

Na obrázku 2.20(vlevo) je zobrazen objekt A, který bude voxelizován za účelem získání objemové reprezentace ze znalosti jeho povrchové reprezentace. Na obrázku 2.20(vpravo) je zobrazen ořezový prostor, který je označen jako objekt B. Ořezovým prostorem je pro použitou ortogonální projekci krychle, která bude mít souřadnice $[-1, -1, 1]$ pro levý dolní roh přední stěny, délka hrany bude rovna 2 jednotkám a krychle bude osově zarovnaná s bázovými vektory globálního souřadnicového systému. Pro voxelizaci je nutné použít paralelní projekci, aby nedocházelo k perspektivnímu zkreslení zobrazeného 3D objektu s rostoucí vzdáleností od projekční roviny. Pro účel voxelizace 3D modelu, je potřeba zarovnat objekt do iniciální polohy, která je stejná pro všechny součásti soustavy. Transformace, které provedou požadované zarovnání 3D objektu do iniciální polohy jsou následující:

1. posunout počátek lokálního souřadnicového systému objektu A do počátku lokálního souřadnicového systému objektu B,

Obrázek 2.21: Voxelizovaný objekt nesmí mít neuzavřený povrch (vlevo) a ani nesmí obsahovat vnitřní příčky (vpravo), aby princip voxelizace pomocí bitových operátorů fungoval správně. Zdroj obrázku [15].

2. zarovnat osu y_A s rovinou $y_B z_B$,
3. zarovnat osu y_A s osou y_B,
4. zarovnat osu x_A s osou x_B,
5. změnit měřítko (zvětšení/zmenšení) objektu A tak, aby objekt A po změně své velikosti ležel uvnitř objektu B a aby co nejlépe vyplňoval prostor definovaný objektem B.

Omezení realizované metody voxelizace objektu s použitím bitových operátorů

Tato metoda má omezení na tvar voxelizovaného 3D objektu, která jsou zobrazena na obrázku 2.21. Tyto omezení plynou z použití bitového operátoru bitové nonekvivalence. Objekt, který bude voxelizován s použitím bitových operátorů, nesmí obsahovat uvnitř svého objemu žádné přepážky a jeho povrch musí být uzavřený (z angl. watertight). Aby bylo možné provést voxelizaci objektu s použitím této metody, je kladen požadavek na grafický procesor, aby podporoval celočíselné textury a bitové operátory.

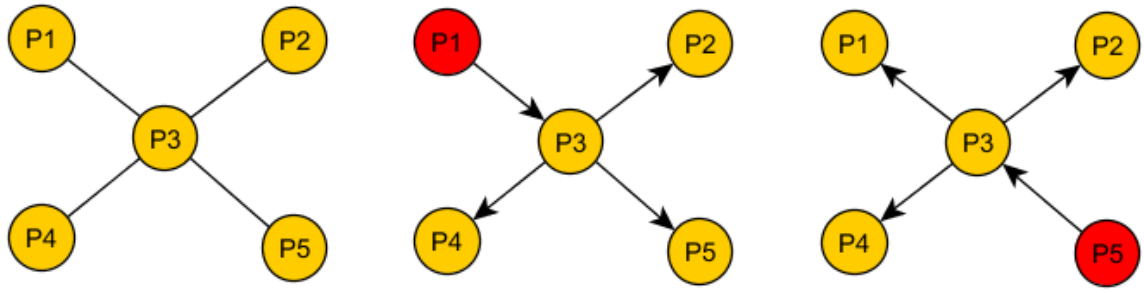
2.3 Vytvoření grafu interakce

Graf interakce je neorientovaný graf, který je tvořen množinou vrcholů a množinou hran. Vrchol grafu interakce představuje mechanickou součást, která obsahuje informaci o svých vlastnostech, a hrana grafu představuje neuspořádanou dvojici vrcholů. Hrana grafu interakce spojuje takovou dvojici součástí, které spolu mají kontakt, a obsahuje informaci o typu jejich spojení. Existence kontaktu dvojice součástí znamená, že změna polohy jedné mechanické součásti způsobí změnu polohy druhé mechanické součásti. Typ spojení součástí se stanoví podle dále popsaného principu v této podkapitole a slouží pro určení, jakým směrem se bude otáčet součást okolo své osy rotace. Pro graf interakce platí, že neexistují v grafu smyčky, které by spojovaly vrchol se sebou samým, a že mezi dvojicí vrcholů nevede více než jedna hrana, jinak by se jednalo o multigraf. Příklad grafu interakce soustavy mechanických součástí je zobrazen na obrázku 2.22. Graf interakce slouží pro stanovení, zda lze provést animaci soustavy, a pro vypočtení parametrů pohybu jednotlivých součástí soustavy na základě zvolené řídící součásti. Pro stanovení, jak se bude šířit kinetická energie udělená

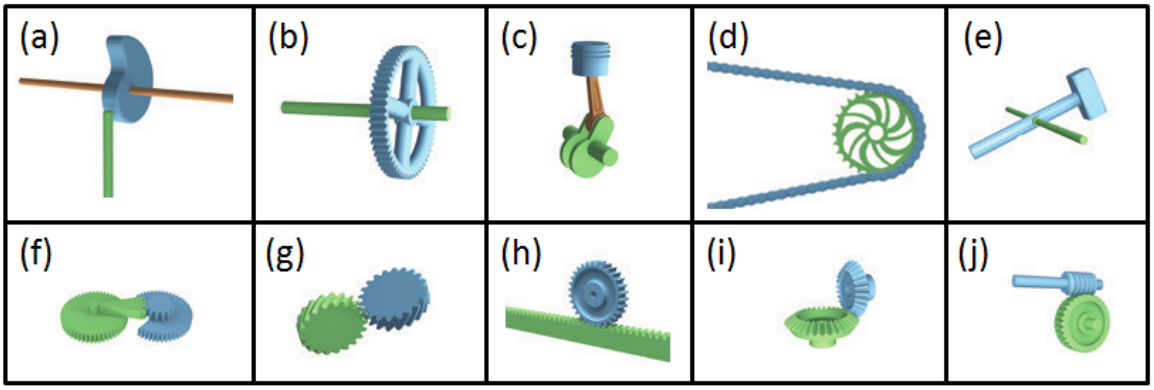
Obrázek 2.22: (vlevo) Soustava mechanických součástí, pro kterou se vytvoří graf interakce (vpravo). V grafu interakce je uzlem grafu mechanická součást a hrana grafu přestavuje existující kolizi mezi dvojicí součástí. Texty obrázku byly přeloženy do češtiny autorkou práce. Zdroj obrázku [27].

řídící součásti analyzovanou soustavou mechanických součástí, je potřeba vědět, jaké součásti spolu mají kontakt (viz podkapitola 2.2). Chování součásti v soustavě závisí na jejích vlastnostech (počet zubů, tvar, rotační osa, poloměry podstav a výška součásti), které byly stanoveny podle principů, které jsou uvedeny v podkapitole 2.1. Pokud bylo určeno, jaká součást soustavy je řídící součástí, pak pro stanovení parametrů animace součástí soustavy dojde ke vnímání grafu interakce jako orientovaného grafu, pro který jsou orientovány hrany ve směru průchodu grafu od řídící součásti (viz obrázek 2.23).

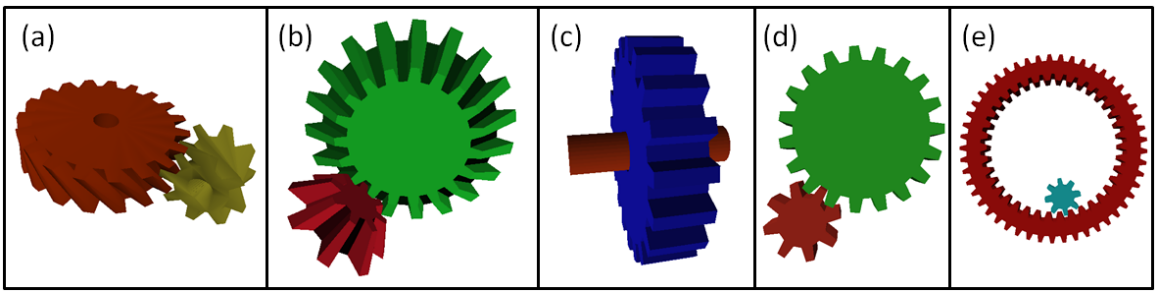
Mitra a kol. [27] prezentují typy spojení součástí, jimiž je vačka (z angl. cam), hřídel (z angl. axle), klika (z angl. crank), řemen (z angl. belt), páka (z angl. lever), převodní stupeň s proměnnou rychlostí (z angl. variable speed gear), spirálové ozubení soukolí (z angl. helical gear), ozubnice – ozubené kolo (z angl. rack – pinion), kuželové ozubené soukolí (z angl. bevel gear) a šnekové soukolí (z angl. worm gear), které jsou znázorněny na obrázku 2.24. Mitra a kol. [27] také podporují základní spojení součástí, kterými jsou válec ve válci a válec na válci. Typ spojení hřídel představuje dvě válcové součásti, které mají koaxiální osy rotace. Typy spojení vačka, klika, řemen, ozubnice – ozubené kolo a páka nejsou v této práci podporovány, protože tvar obou součástí tvořící uvedená spojení není buď válec a nebo komolý kužel. Typ spojení převodní stupeň s proměnnou rychlostí není v této práci podporován, protože pro tento druh spojení by bylo potřeba vytvořit časově proměnných graf interakce z důvodu měnící se rychlosti řízené součásti při konstantní rychlosti řídící součásti. Typ spojení šnekové soukolí není v této práci podporován, protože nejsme schopni rozlišit šnekové soukolí a námi podporované spojení spirálové ozubené soukolí z důvodu, že obě dvě spojení tvoří součásti tvaru válce, které mají kolmé osy rotace. Množina podporovaných typů spojení dvojice součástí v této práci je znázorněna na obrázku 2.25 a tvoří podmnožinu typů spojení, které prezentují Mitra a kol. [27]. Typ spojení dvou součástí, které spolu mají kontakt, se určí podle tvaru součástí a vzájemné polohy jejich rotačních os. Vzájemná poloha rotačních os závisí na velikosti úhlu, který svírají směrové vektory os rotace analyzované



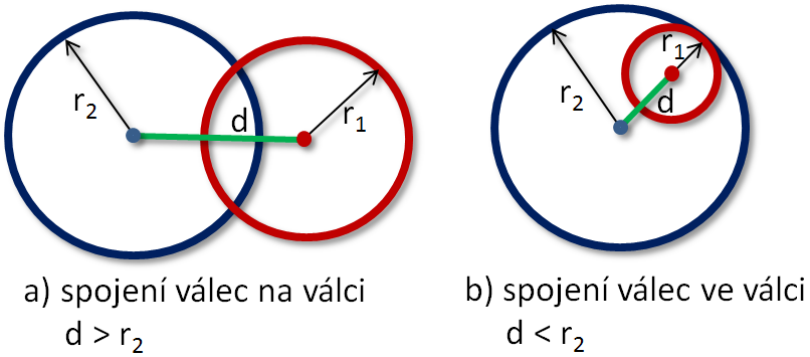
Obrázek 2.23: (vlevo) Neorientovaný graf interakce. (uprostřed) Vnímání orientace hran grafu interakce pro řídící součást P_1. (vpravo) Vnímání orientace hran grafu interakce pro řídící součást P_5.



Obrázek 2.24: Typy spojení součástí: (a) vačka, (b) hřídel, (c) klika, (d) řemen, (e) páka, (f) převodní stupeň s proměnnou rychlostí, (g) spirálové ozubení soukolí, (h) ozubnice – ozubené kolo, (i) kuželové ozubené soukolí a (j) šnekové soukolí. Zdroj obrázku [27].



Obrázek 2.25: Typy spojení součástí realizované v této práci: (a) spirálové ozubené soukolí, (b) kuželové ozubené soukolí, (c) hřídel, (d) válec na válci, (e) válec ve válci.



Obrázek 2.26: Stanovení typu spojení dvojice součástí, které mají rovnoběžné osy symetrie. Podle vzdálenosti os a největšího poloměru podstavy součásti dojde ke stanovení, zda se jedná o typ spojení (a) nebo (b).

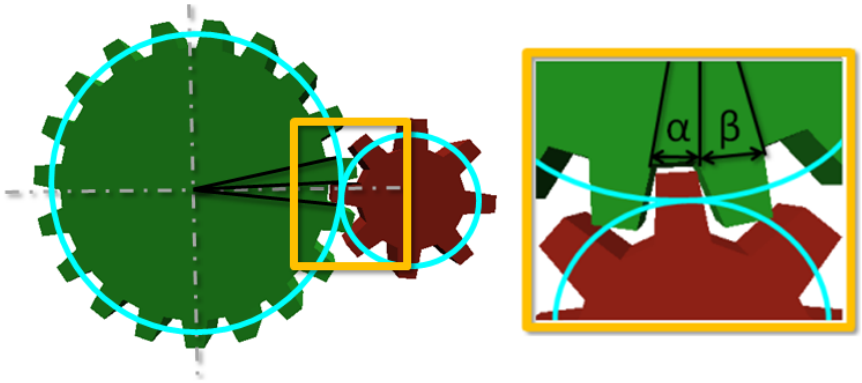
dvojice součástí. Úhel mezi dvěma vektory u, v je takové číslo $\alpha \in < 0, 2\pi)$, pro které platí

$$cos\alpha = \frac{u \cdot v}{|u||v|}. \tag{2.6}$$

Přímky jsou na sebe kolmé, pokud skalárních součin jejich směrových vektorů je roven 0. Pro rovnoběžné přímky platí, že skalární součin jejich směrových vektorů je roven 1 a směrový vektor jedné přímky je násobkem směrového vektoru druhé přímky. Koaxiální (totožné) přímky jsou speciálním případem rovnoběžných přímek, které mají nulovou vzdálenost.

1. kolmé osy
   (a) spirálové ozubené soukolí, pokud tvar obou dvou součástí je válec
   (b) kuželové ozubené soukolí, pokud tvar obou dvou součástí je kužel
2. rovnoběžné osy
   (a) válec na válci
   (b) válec ve válci
3. totožné osy
   (a) hřídel

Pokud jsou rotační osy dvojice součástí, které spolu mají kontakt, rovnoběžné a tvary součástí jsou válce, pak je potřeba, aby bylo možné rozhodnout, zda se jedná o typ spojení válec ve válci nebo válec na válci, vypočítat vzdálenost rotačních os obou součástí. Dále popisovaný princip určení, zda se jedná o typ spojení válec ve válci nebo válec na válci, je znázorněn na obrázku 2.26. Nechť r_1 je poloměr první součásti, r_2 je poloměr druhé součásti a vzdálenost jejich rotačních os je d. Určí se vzdálenost projekcí počátků lokálních souřadnicových systémů obou dvou součástí do roviny, v níž leží podstava součásti, která má větší poloměr. Pokud je vypočtená vzdálenost menší nebo rovna poloměru podstavy součásti,



Obrázek 2.27: (vlevo) Tvar zubů pro kolidující součásti soustavy musí být stejný. (vpravo) Detail zubů. Modrozelená barva – roztečné kružnice součástí.

která má z vyšetřované dvojice součástí větší poloměr, pak se jedná o spojení válec ve válci. V opačném případě se jedná o spojení součástí typu válec na válci.

Pro stanovení parametrů animace soustavy mechanických součástí je potřeba, aby dvojice součástí, které spolu mají kontakt, měla stejný tvar zubů. Tento požadavek je kladen na analyzovanou soustavu z důvodu, aby byla animace pohybu soustavy součástí věrohodná. Aby bylo možné použít při výpočtu parametrů animace součásti soustavy buď počet zubů a nebo poloměr roztečné kružnice, tak je potřeba zajistit, aby byla tloušťka zubu na roztečné kružnici rovna velikosti mezery mezi sousední dvojicí zubů součásti. Roztečná kružnice je taková kružnice, která je soustředná se středem kruhové podstavy součásti a která má poloměr roven vzdálenosti od jejího středu k bodu, který leží v poloviční vzdálenosti zapuštění zubů dvojice součástí, které spolu mají kontakt, ve směru výšky zubu (viz obrázek 2.27(vlevo) modrozelená kružnice). Velikost mezery zubu je rovna délce oblouku, který je vymezen na roztečné kružnici pro dvojici sousedních zubů. Tloušťka zubu je rovna délce oblouku na roztečné kružnici, kde se vyskytuje zub. Na obrázku 2.27 je zobrazena vlevo soustava dvou mechanických součástí a vpravo je zobrazen detail jejich zubů. Počet zubů pro každou součást musí být celé číslo, tzn. musí platit, že $360/(\alpha + \beta)$ je celočíselný podíl beze zbytku, kde úhly α, β jsou vyjádřeny ve stupních. Písmeno řecké abecedy α představuje v uvedeném vzorci středový úhel kruhové výseče, který vymezuje šířku mezery mezi sousedními zuby a β je středový úhel kruhové výseče, který určuje tloušťku zubu (viz obrázek 2.27(vpravo)).

Protože použitá metoda v této práci, která stanovila lokální souřadnicový systém součásti, vypočetla neorientované osy, je potřeba určit orientaci takové osy lokálního souřadnicového systému součásti soustavy, která představuje osu rotace součásti. Vypočtením orientace rotační osy součásti se zajistí, že budou rotační osy součástí soustavy, které spolu mají kontakt, vůči sobě správně orientovány. Správnou orientací rotačních os součástí, které spolu mají kontakt, rozumíme skutečnost, že se tyto součásti soustavy budou otáčet vůči sobě správným směrem. Knihovna OpenGL pracuje nad pravotočivou souřadnou soustavu, a proto je potřeba zajistit, že lokální souřadnicový systém každé mechanické součásti bude pravotočivý. Nyní bude vysvětlen princip nalezení správné orientace rotačních os součástí soustavy, který je znázorněn na obrázku 2.28, kde součást P_1 představuje řídící součást a součást P_2 představuje řízenou součást. Pro stanovení správné orientace rotačních os součástí dochází

k průchodu grafu interakce směrem od řídící součásti a analyzování každé jeho hrany. Orientace rotační osy řídící součásti se bere jako referenční hodnota a dochází k případné změně orientace rotační osy řízené součásti aktuálně analyzované hrany. Stanovení, zda je osa rotace řízené součásti správně orientována vůči ose rotace řídící součásti, se provede na základě dále popsaných principů podle typu spojení analyzované dvojice součástí, mezi nimiž existuje hrana v grafu interakce.

Orientace os rotace pro spojení válec na válci, válec ve válci a osa. Pro směrové vektory rotačních os $\vec{a_1}$, $\vec{a_2}$ a pro směrový vektor $\vec{b_1}$, který prochází počátky lokálních souřadnicových systémů součástí P_1 a P_2, musí platit, že $\vec{b_1} \times \vec{a_1} = \vec{b_1} \times \vec{a_2}$ (viz obrázek 2.28(a)).

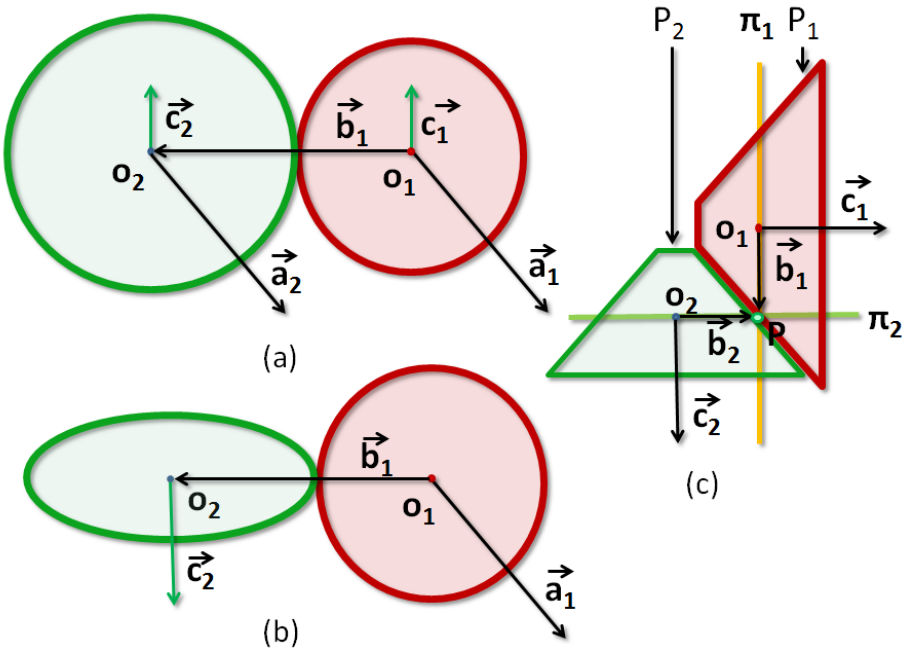
Orientace os rotace pro spojení spirálové ozubené soukolí. Pro směrové vektory rotačních os $\vec{a_1}$, $\vec{c_2}$ a pro směrový vektor $\vec{b_1}$, který prochází počátky lokálních souřadnicových systémů součástí P_1 a P_2, musí platit, že $\vec{c_2} \times \vec{b_1} = -\vec{a_1}$ (viz obrázek 2.28(b)).

Orientace os rotace pro spojení kuželové ozubené soukolí. Princip stanovení, zda je osa rotace řízené součásti P_2 správně orientována vůči ose rotace řídící součásti P_1, je znázorněn na obrázku 2.28(c). Nechť bod o_i představuje počátek lokálního souřadnicového systému součásti P_i. Rovina π_i je taková rovina, jejíž normálový vektor je shodný se směrovým vektorem osy rotace $\vec{c_i}$ součásti P_i, a v níž leží bod o_i, kde $i \in \{1, 2\}$. Nalezne se bod P jako kolmá projekce o_2 do roviny π_1. Vytvoří se vektory $\vec{b_1}$, $\vec{b_2}$ směřující od o_i k bodu P. Pokud mají směrové vektory os rotace obou součástí zároveň stejný nebo opačný směr jako vektory $\vec{b_i}$, kde $i = 1$ pro P_2 a $i = 2$ pro P_1, pak je osa rotace součásti P_2 správně orientována.

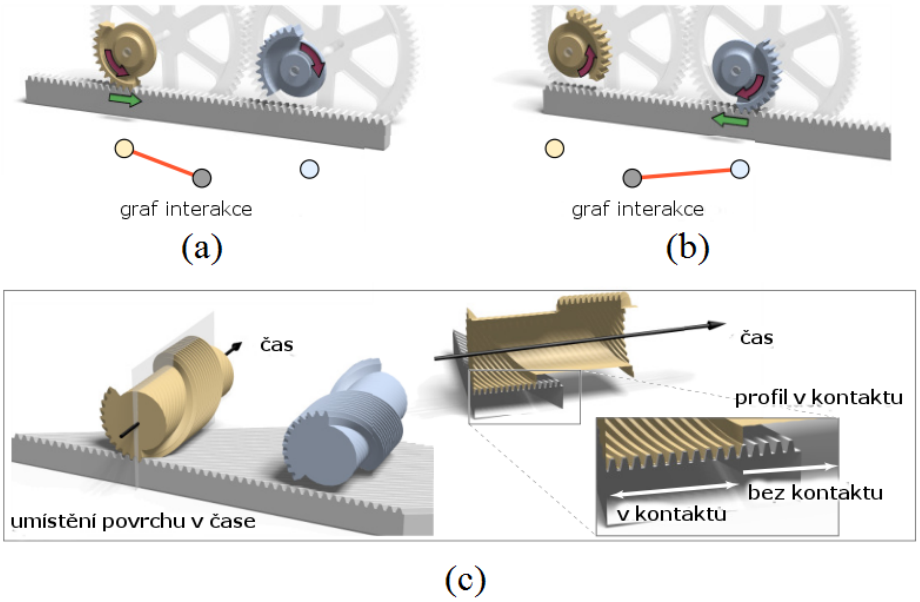
Omezení grafu interakce Problematika výpočtů parametrů animace součástí by se zkomplikovala, pokud by docházelo například k tomu, že by se během životního cyklu pohybu některé součásti měnila množina objektů, s nimiž by daná součást měla kontakt. Mitra a kol. [27] řeší tuto problematiku vytvořením časově proměnného grafu interakce (viz obrázek 2.29). Časově proměnný graf interakce se liší od grafu interakce realizovaného v této práci v tom, že se mění množina hran grafu v závislosti na čase. V této práci se parametry animace součásti a seznam součástí, s nimiž má součást kontakt, po vytvoření grafu interakce nesmí měnit.

2.4 Dynamická vizualizace soustavy mechanických součástí

Dynamická vizualizace znázorňuje, jak se bude šířit kinetická energie udělená řídící součásti uvnitř soustavy mechanických součástí. Kinetická energie, která byla udělena řídící součásti, se soustavou šíří tak, že pohyb řídící součásti vyvolá pohyb součástí, které s ní mají kontakt. Princip šíření kinetické energie soustavou mechanických součástí tak představuje jakýsi řetězový efekt, kde přenesení kinetické energie na součást P_i vyvolá pohyb součásti P_j, která má kontakt se součástí P_i. Součást P_j, která byla rozpohybována součástí P_i, pak dále předá svojí kinetickou energii součástem, které s ní mají kontakt, kromě součásti P_i. Ze znalosti součástí soustavy, které spolu mají kontakt, se určí, poloha kterých součástí v soustavě se



Obrázek 2.28: Určení orientace rotační osy podle typu spojení: (a) válec na válci, válec ve válci, osa, (b) spirálové ozubené soukolí a (c) kuželové ozubené soukolí.



Obrázek 2.29: Časově proměnný graf interakce: (a) Ve vyobrazený časový okamžik existuje kolize hnědé součásti s ozubnicí. (b) V jiný časový okamžik nežli v (a) existuje kolize šedé součásti s ozubnicí. (c) Analýza povrchů součástí soustavy určuje, kdy dojde ke kolizi součástí soustavy. Texty obrázku byly přeloženy do češtiny autorkou práce. Zdroj obrázku [27].

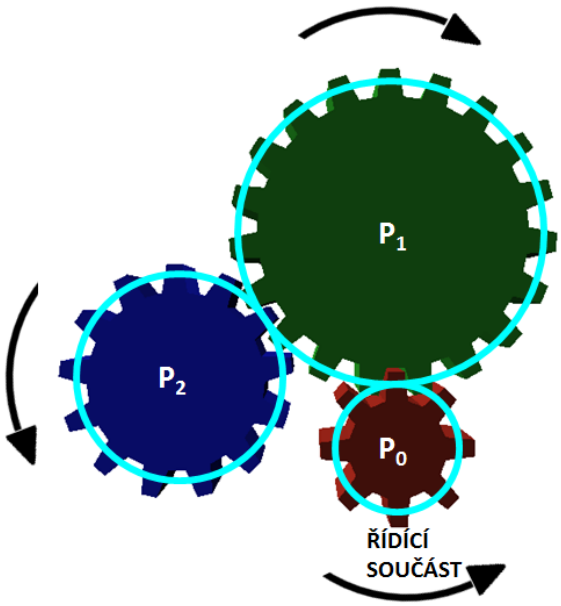
změní v závislosti na změně polohy součásti, která byla rozpohybována. Podle vlastností každé součásti soustavy se stanoví, jaký druh (např. otáčivý pohyb pro rotační tělesa) a velikost pohybu bude součást vykonávat v analyzované soustavě. V této práci jsou podporována pouze rotační tělesa a to konkrétně válec a komolý kužel. Aby bylo možné vizualizovat rotační pohyb součásti, je potřeba znát její osu rotace, okolo které se bude otáčet. V případě válce a komolého kužele je osa rotace zároveň jejich osou symetrie. Rychlost otáčení součásti závisí na rychlosti, která byla udělena řídící součásti, a na velikosti součásti.

Většina 3D modelů, vytvořená například programem AutoCAD[4], neobsahuje informaci o tom, jaký vztah mají navzájem vůči sobě mechanické součásti uvnitř zobrazované soustavy. Sice existuje řada programů, které mají pomoci uživateli animovat mechanické součásti (např. SolidWorks Motion [32]), ale bohužel stále je potřeba nezanedbatelná interakce uživatele. Pro automatizaci procesu animace pohybu mechanických součástí pro danou soustavu a minimalizaci potřebné interakce uživatele je nejprve potřeba stanovit vlastnosti součásti například na základě její geometrie pro zjištění, jak se bude pohybovat součást v soustavě. Dále je potřeba určit vzájemné působení mechanických součástí uvnitř dané soustavy na základě znalosti kolidujících součástí. Tyto dvě získané informace jsou spojeny dohromady v grafu interakce, jehož princip vytvoření byl vysvětlen v podkapitole 2.3.

V následujícím textu bude nejprve vysvětleno, jak se vypočtou rychlosti rotace součástí soustavy, která neobsahuje nepohyblivou součást. Poté bude vysvětleno, jakým způsobem je možné nalézt poloměry roztečných kružnic součástí soustavy, které mohou sloužit pro vypočtení rychlosti rotace součásti. V této práci je realizováno stanovení rychlosti rotace součástí ze znalosti počtu zubů součásti. Dále bude uvedeno, jakým způsobem se určí směr rotace součásti. Poté budou definovány podporované typy součástí v této práci. Bude vysvětlen princip výpočtu rychlosti rotace součástí soustavy, která obsahuje alespoň jednu nepohyblivou součást. Pro soustavu, která obsahuje alespoň jednu nepohyblivou součást, pak bude analyzováno, zda dojde a popřípadě podle jakých principů k rozkladu grafu interakce na podgrafy. Bude stanoveno, zda je animace soustavy součástí proveditelná. Dále bude diskutována skutečnost možnosti vzniku aliasingu v časové oblasti při dynamické vizualizaci soustavy mechanických součástí. V neposlední řadě budou uvedeny omezení této realizované metody pro stanovení dynamické vizualizace soustavy mechanických součástí.

2.4.1 Výpočet rychlosti rotace součástí soustavy, která neobsahuje nepohyblivou součást

Předpokládejme, že v soustavě mechanických součástí existuje řídící součást, která je připojena k motoru, jehož rychlost a směr rotace lze měnit. S řídící součástí mají kontakt některé mechanické součásti, které tvoří analyzovanou soustavu. Pro vytvoření dynamické animace soustavy je potřeba vědět, jakou rychlost, směr rotace a podle jaké osy rotace se bude otáčet každá součást analyzované soustavy na základě zadaných parametrů pohybu řídící součásti. Situace jednoduché soustavy mechanických součástí, kterou tvoří tři součásti, je zobrazena na obrázku 2.30. Na obrázku si můžeme všimnout, že díky skutečnosti, že součásti analyzované soustavy jsou ozubená kola a zuby každé kolidující dvojice součástí jsou do sebe zaklesnuty, tak otočení jedné součásti způsobí otočení součásti, která s ní má kontakt. Velikost ozubeného kola lze vyjádřit buď pomocí poloměru nebo pomocí počtu zubů. Tyto dvě možnosti pro určení velikosti ozubeného kola jsou ekvivalentní, pokud je známa

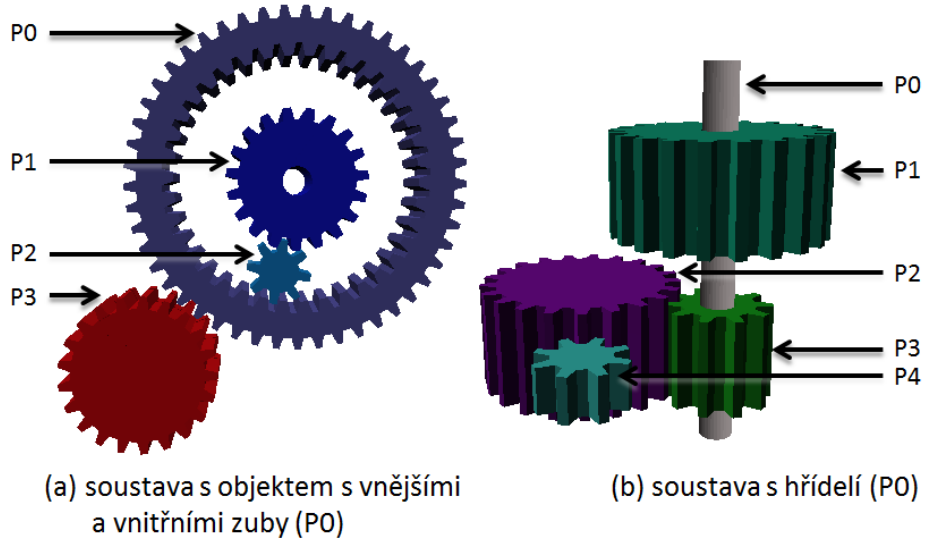


Obrázek 2.30: Znázornění jednoduché soustavy mechanických součástí. Roztečná kružnice každé součásti je označena modrozelenou barvou.

tloušťka zubu a mezera mezi dvojicí sousedních zubů součásti. Při zvolení přístupu výpočtu parametrů animace součásti ze znalosti poloměru součásti je potřeba uvažovat na místo poloměru podstavy součásti poloměr roztečné kružnice (z angl. pitch circle), která je zobrazena na obrázku 2.30 modrozelenou barvou. Na obrázku je vidět, že roztečné kružnice dvojice součástí, které spolu mají kontakt, se protínají v jednom bodě. Aby pohyb součástí zůstal synchronizován, pak musí platit pro každou dvojici sousedních součástí, že velikosti jejich obvodových rychlostí pro body, které leží na roztečných kružnicích, jsou stejné. Počet zubů je přímo úměrný poloměru roztečné kružnice. Velikost rozteče součásti, která představuje vzdálenost dvou shodných bodů, které leží na dvou sousedících zubech součásti, je rovna dvojnásobku tloušťky zubu a lze také vyjádřit jako podíl obvodu roztečné kružnice počtem jejích zubů. Velikost rozteče lze také definovat jako součet tloušťky zubu a velikosti mezery mezi dvojicí sousedních zubů na roztečné kružnici. V případě, že velikost rozteče je rovna dvojnásobku tloušťky zubu a že velikost a tvar zubu je stejný pro součásti, které spolu mají kontakt, lze v dále uvedených vzorcích pro výpočet rychlosti animace součásti beztrestně zaměnit poloměr roztečné kružnice za počet zubů součásti.

Výpočet parametrů animace součástí soustavy, který je realizován v této práci, je založen na principu metod, které popisuje Milton [25]. Poměr rychlostí dvou součástí soustavy, které spolu mají kontakt, je roven $R = \omega_A/\omega_B = r_B/r_A = N_B/N_A$, kde N je počet zubů, r je poloměr roztečné kružnice a ω je úhlová rychlost součásti. Čím větší má součást poloměr, tím pomaleji se otáčí. Velikost obvodové rychlosti v je rovna $v = \omega \cdot r$ popř. $v = \omega \cdot N$, kde ω je úhlová rychlost, r je poloměr roztečné kružnice a N je počet zubů součásti.

Cesta v grafu interakce představuje posloupnost vrcholů, pro kterou platí, že existuje hrana z výchozího vrcholu do jeho následovníka. Zároveň pro cestu grafu musí platit, že



Obrázek 2.31: Příklad jednoduchých soustav, které obsahují (a) součást s vnějšími a vnitřními zuby, (b) hřídel.

se na ní neopakuje žádný vrchol a ani hrana. Cesta v grafu interakce reprezentuje řetězec součástí, které spolu mají kontakt. Milton [25] říká, že pokud neexistuje v soustavě na cestě mezi řídící součástí a analyzovanou součástí hřídel nebo součást, která má vnitřní a vnější zuby (viz obrázek 2.25(e) hnědá součást) a která na této cestě má zároveň spojení válec na válci a válec ve válci se sousedními součástmi, pak rychlost libovolné analyzované součásti na této cestě v soustavě závisí pouze na počtu zubů řídící součásti a aktuálně analyzované součásti. Ukažme si na jednoduchém příkladě přenosového řetězce, který je zobrazen na obrázku 2.30, že tomu tak skutečně je. Řídící součástí je součást P_0 a aktuálně analyzovaná součást je součást P_2. Rychlost rotace ω_{P_2} součásti P_2 je rovna $\omega_{P_0} \cdot N_{P_0}/N_{P_2}$. Pokud by se vypočítala rychlost otáčení pro součást P_2 ze znalosti počtu zubů součásti P_1, pak by výpočet vypadal následovně $\omega_{P_0}/\omega_{P_1} = N_{P_1}/N_{P_0}, \omega_{P_1}/\omega_{P_2} = N_{P_2}/N_{P_1}$. Poté by se vynásobily stanovené poměry rychlostí a výsledný výraz by vypadal následovně $\omega_{P_0}/\omega_{P_2} = N_{P_2}/N_{P_0}$. Z uvedeného příkladu výpočtu úhlové rychlosti vyšetřované součásti vyplývá, že v případě, že v soustavě na analyzované cestě mezi řídící součástí a vyšetřovanou součástí neexistuje hřídel a ani součást, která má vnitřní a vnější zuby a pro níž existuje na této cestě spojení válec ve válci a válec na válci, pak nezáleží na tom, zda se použije pro výpočet rychlosti pohybu vyšetřované součásti pouze počet zubů řídící součásti a aktuálně analyzované součásti, a nebo pokud se zahrnou do výpočtu rychlosti analyzované součásti i součásti, které se vyskytují na cestě mezi analyzovanou součástí a řídící součástí.

Milton [25] tvrdí, že pokud existuje v soustavě na cestě mezi řídící součástí a analyzovanou součástí součást P_i, která má vnitřní (směřující dovnitř součásti) a vnější (směřující vně součásti) zuby (viz obrázek 2.25(e) hnědá součást) a na analyzované cestě existuje typ spojení pro tuto součást válec na válci i válec ve válci, pak je potřeba při stanovení úhlové rychlosti analyzované součásti vzít v úvahu i vlastnosti (počet zubů) součástí, které kolidují se součástí P_i na nalezené cestě, a počet vnitřních a vnějších zubů součásti P_i. Uvažujme soustavu

zobrazenou na obrázku 2.31(a), která obsahuje součást $P0$, která má vnitřní a vnější zuby, a pro kterou existuje na nalezené cestě zároveň spojení válec ve válci ($P0$ spojena se součástí $P2$) a spojení válec na válci ($P0$ spojena se součástí $P3$). Nechť je řídící součástí součást $P1$ a analyzovanou součástí je součást $P3$. Na nalezené cestě mezi řídící součástí a analyzovanou součástí jsou součásti $P2$ a $P0$. Rychlost analyzované součásti $P3$ nelze vypočítat pouze na základě znalosti vlastností řídící součásti a analyzované součásti, protože obvodové rychlosti součásti $P0$ na jejím vnitřním a vnějším poloměru jsou odlišné. Rychlost součásti $P3$ je rovna $\omega_{P1} \cdot (N_{P1} \cdot N_{P0_{outer}})/(N_{P0_{inner}} \cdot N_{P3})$, kde N_{P_i} je počet zubů součásti Pi, $N_{P0_{outer}}$ je počet vnějších zubů součásti $P0$ a $N_{P0_{inner}}$ je počet vnitřních zubů součásti $P0$.

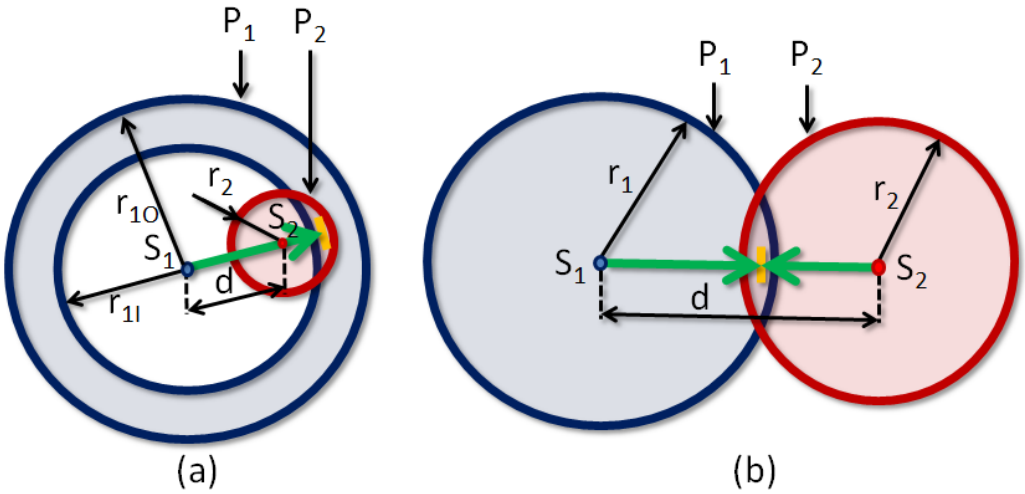
Pokud existuje v soustavě na cestě mezi řídící součástí a analyzovanou součástí hřídel, pak tato hřídel vykazuje dvě odlišná chování dle její funkce v soustavě. Hřídel má funkci v soustavě buď jako podpěra pro ozubená kola, která jsou na ní nasazena, a nebo se hřídel pohybuje stejným směrem a úhlovou rychlostí jako součást, která je na ní nasazena a která má nenulovou rychlost rotace. Pokud slouží hřídel jako podpěra pro ozubená kola, která jsou na ní nasazena, pak se hřídel nepohybuje, tzn. má nulovou kinetickou energii, a proto nešíří pohyb soustavou. Pokud je na hřídeli nasazena součást P_i, která má nenulovou rychlost rotace, zároveň je na ní nasazena nepohyblivá (fixní) součást P_j a dochází k šíření kinetické energie směrem od součásti P_i, pak se bude hřídel otáčet stejnou úhlovou rychlostí jako součást P_i a zároveň se bude protáčet na nepohyblivé součásti P_j. Hřídel má tvar válce a není ozubenou součástí. Obvodová rychlost hřídele a obvodová rychlost ozubeného kola, které je nasazeno na hřídeli, není stejná, a proto pro určení rychlosti analyzované součásti musí být uvažovány i součásti na cestě mezi řídící součástí a analyzovanou součástí, které mají kontakt s hřídelí. Příklad soustavy s hřídelí je zobrazen na obrázku 2.31(b). Uvažujme, že řídící součástí v soustavě 2.31(b) je součást $P1$ a analyzovanou součástí je součást $P4$. Úhlová rychlost součásti $P4$ je rovna $\omega_{P1} \cdot (N_{P1} \cdot \ N_{P3})/(N_{P1} \cdot \ N_{P4})$.

2.4.2 Stanovení poloměru roztečné kružnice součásti

Pro stanovení poloměru roztečných kružnic dvojice součástí, které spolu mají kontakt, je potřeba nalézt bod, který leží v poloviční vzdálenosti výšky zaklesnutí zubů dvojice součástí. Pro bod, který leží na roztečné kružnici platí, že obvodová rychlost tohoto bodu je stejná pro dvojici součástí, které spolu mají kontakt. Výpočet poloměru roztečných kružnic dvojice kolidujících součástí závisí na typu jejich spojení. Nechť existuje kontakt součásti P_1, P_2, pak vzdálenost počátků lokálních souřadnicových systémů těchto součástí je označena jako d. Pokud mají součásti pouze vnější zuby, pak poloměry těchto součástí jsou označeny jako r_1, r_2. Poloměr roztečné kružnice součásti P_i je označen jako r_{iP}. Pokud existuje pro součást P_i typ spojení válec ve válci, pak její vnější poloměr je označen jako r_{iO}, její vnitřní poloměr je označen jako r_{iI} a proměnná označená ve vzorci 2.9 jako x představuje výšku zaklesnutí zubů součásti P_1 a P_2. Pokud je typ spojení analyzované dvojice součástí válec na válci, pak se určí poloměr roztečných kružnic dvojice součástí dle vzorce 2.7 a 2.8, a pokud je typ spojení válec ve válci, pak se poloměr roztečných kružnic součástí určí podle vzorce 2.10 a 2.11. Princip nalezení poloměru roztečných kružnic pro dvojici kolidujících součástí pro spojení válec na válci a válec ve válci je znázorněn na obrázku 2.32.

$$r_{1P} = r_1 - (r_1 + r_2 - d) \cdot 0.5 \tag{2.7}$$

$$r_{2P} = r_2 - (r_1 + r_2 - d) \cdot 0.5 \tag{2.8}$$



Obrázek 2.32: Princip určení poloměrů roztečných kružnic kolidující dvojice součástí, které mají tvar válce pro spojení (a) válec ve válci a (b) válec na válci.

$$x = d + r_2 - r_{1I} \tag{2.9}$$

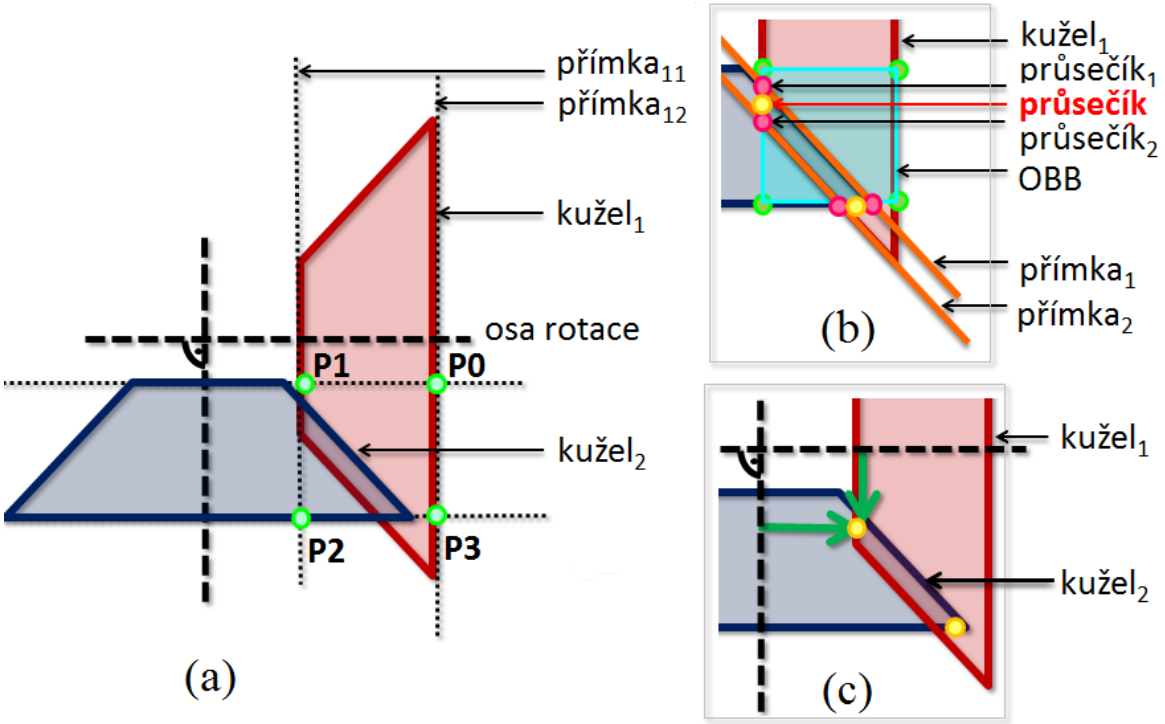
$$r_{1P} = r_{1I} + x/2 \tag{2.10}$$

$$r_{2P} = r_2 - x/2 \tag{2.11}$$

Princip nalezení poloměru roztečných kružnic pro dvojici součástí, které spolu mají kontakt a které mají tvar komolého kužele, je zobrazen na obrázku 2.33. Pro určení poloměrů roztečných kružnic je potřeba nalézt bod, který leží v poloviční vzdálenosti výšky zapuštění zubů dvojice kolidujících součástí. Jeden z bodů, který tuto podmínku splňuje je na obrázku 2.33(b) označen jako průsečík a je zvýrazněn žlutou barvou. Poloměr roztečné kružnice komolého kužele je závislý na zvolené výšce podél rotační osy součásti. Kolmá vzdálenost tohoto průsečíku k ose rotace součásti odpovídá poloměru roztečné kružnice součásti, která byla získána ve zvolené vzdálenosti mezi podstavami komolého kužele. Tento bod se nalezne tak, že se vytvoří přímka pro každou podstavu obou dvou komolých kuželů, která vznikne jako průsečík roviny, ve které leží podstava komolého kužele, s rovinou, jejíž směrové vektory jsou totožné s rotačními osami obou dvou objektů. Určí se průsečíky vytvořených čtyř přímek (viz obrázek 2.33(a)), které leží ve vrcholech čtyřúhelníku. Poté se vytvoří přímky, které jsou označené na obrázku 2.33(b) jako *přímka*$_1$, *přímka*$_2$. Naleznou se průsečíky těchto dvou přímek se čtyřúhelníkem (viz na obrázku 2.33(b) růžové body). Vybere se jeden pár průsečíků a stanoví se bod, který leží v jejich poloviční vzdálenosti. Tento bod je označen jako *průsečík* a je hledaným průsečíkem. Poloměr roztečné kružnice součásti je roven kolmé vzdálenosti nalezeného bodu označeného jako *průsečík* k ose rotace součásti.

2.4.3 Stanovení směru rotace součásti

Typ spojení dvojice součástí, který představuje hranu grafu interakce mezi příslušnou dvojicí součástí, určuje, jaký směr rotace bude mít řízená součást vzhledem ke směru rotace řídící



Obrázek 2.33: Projekce konvexních obálek součástí, které mají tvar komolého kužele. (a) Nalezení OBB, která je tvořena bodem $P0$, $P1$, $P2$ a $P3$. (b) Nalezení průsečíků přímky$_1$ a přímky$_2$ s OBB. (c) Nalezení poloměrů roztečných kružnic součástí.

součásti. V této práci jsou definovány dva směry rotace součásti. Součást se otáčí okolo své osy rotace buď po směru hodinových ručiček (CW) a nebo proti směru hodinových ručiček (CCW). Nechť analyzovaná hrana grafu interakce spojuje součásti P_i, P_j, směr rotace řídící součásti P_i je označen jako D_i a směr rotace řízené součásti P_j je označen jako D_j. Určení směru rotace součásti P_j se provede podle následujícího vztahu 2.12.

$$D_j = \begin{cases} D_i & \text{pro spojení typu válec ve válci, spirálové ozubené soukolí a hřídel} \\ \neq D_i & \text{pro spojení typu kuželové ozubené soukolí a válec na válci} \end{cases} \tag{2.12}$$

2.4.4 Definice typů součástí

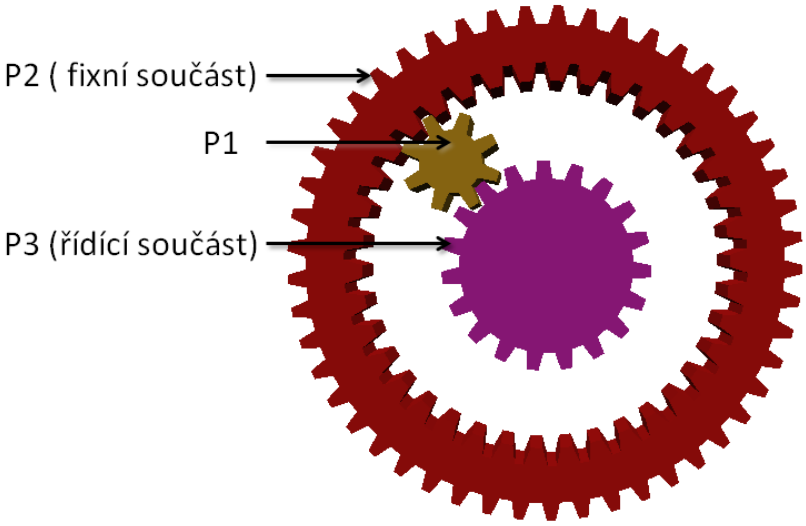
Rozhodnutí, zda je animace soustavy proveditelná, závisí na tom, jaká součást byla vybrána jako řídící součást, jaké mají prostorové uspořádání součásti uvnitř soustavy a jakého typu jsou součásti. Typ součásti určuje, zda a jakým způsobem může součást měnit svojí polohu v prostoru. Pokud součást může měnit svojí polohu v prostoru, pak typ součásti určuje, podle jaké osy rotace bude moct součást rotovat.

V této práci jsou podporovány mechanické součásti následujících typů:

1. statický,
2. dynamický,
3. fixní.

Fixní součást představuje nepohyblivou součást, která nesmí měnit svoje umístění v prostoru a vytváří tak opěrná místa soustavy. Statická součást může rotovat pouze kolem své osy rotace. Pokud je vzájemná poloha rotační osy statické součásti koaxiální s osou rotace fixní součásti, pak je rotace statické součásti proveditelná z důvodu, že jsou jejich osy rotace totožné. Dynamická součást může rotovat jak kolem své osy rotace tak kolem rotační osy fixní součásti. Pokud není dynamická součást přinucena rotovat okolo rotační osy fixní součásti, pak rotuje pouze okolo své rotační osy. Pokud je fixní hřídel, na kterou je nasazeno ozubené kolo, které není fixní součástí, pak toto ozubené kolo se stane statickou součástí, protože jeho pohyb je omezen zafixovanou hřídelí.

Speciálním případem soustavy ozubených kol je planetová převodovka, kterou tvoří jeden či více planetových (či planetárních) převodů. Milton [25] vysvětluje funkci planetové převodky a prezentuje výpočet parametrů pohybu součástí soustavy, kterou tvoří centrální kolo, planety a prstencové kolo (viz obrázek 2.34). Počet jednotlivých druhů součástí, ze kterých je složen planetový převod, může být libovolný. Počet stupňů volnosti systému představuje počet parametrů, které se mohou nezávisle měnit. Počet stupňů volnosti (DOF) planetové převodovky je roven DOF = A + P - C, kde A je počet centrálních kol, prstencových kol a ramen, P počet planet a C je počet párů součástí, které spolu kolidují. Rameno přestavuje tuhou součást, která spojuje planetu a centrální kolo. Počet stupňů volnosti planetové převodovky, která je zobrazena na obrázku 2.34, je roven 2 $(3 + 1 - 2)$. Pokud je součást soustavy nucena rotovat okolo rotační osy fixní součásti, pak tato součást má 2 stupně volnosti (směr rotace okolo rotační osy fixní součásti a směr rotace okolo své osy rotace). Pohyb každé mechanické součásti, která je nucena rotovat okolo rotační osy fixní součásti, je složen z pohybu této součásti okolo vlastní osy rotace a pohybu okolo rotační osy nepohyblivé součásti.



Obrázek 2.34: Planetová převodovka je tvořena planetovým převodem, který je složen z centrálního kola ($P3$), planety $P1$ a prstencového kola ($P2$).

2.4.5 Výpočet rychlosti rotace ramene okolo rotační osy nepohyblivé součásti

Milton [25] prezentuje dva základní postupy pro určení úhlových rychlostí součástí soustavy, ve které existuje alespoň jedna nepohyblivá součást. První metodou je výpočet úhlových rychlostí součástí pomocí tabulky a druhým přístupem je výpočet úhlových rychlostí součástí pomocí algebraického výpočtu (vzorce). Výsledný pohyb každé součásti je popsán rovnicí $\omega_{gear} = \omega_{arm} + \omega_{gear/arm}$, kde ω_{arm} je rotace ramene okolo nepohyblivé součásti a $\omega_{gear/arm}$ představuje rotaci součásti okolo vlastní osy rotace relativně k rotaci ramene. V této práci je realizována metoda výpočtu úhlových rychlostí součástí soustavy pomocí algebraického výpočtu.

Při použití metody využívající tabulky se vytvoří pro každou součást, která tvoří soustavu, jeden řádek v tabulce. V tabulce existují tři sloupce, kde jeden sloupec obsahuje úhlovou rychlost rotace ramene (ω_{arm}), další sloupec obsahuje informaci o relativní rychlosti rotace součásti vůči rotaci ramene ($\omega_{gear/arm}$) a v dalším sloupci je obsažena informace o výsledné úhlové rychlosti součásti (ω_{gear}). Uvažujme, že R představuje poměr počtu zubů řídících a řízených součástí na cestě mezi požadovanou dvojicí součástí soustavy. Tento poměr se stanoví stejným způsobem, který byl popsán v sekci 2.4.1 této podkapitoly. Nejprve se do tabulky doplní známé informace jako je výsledná rychlost řídící součásti a nulová výsledná rychlost nepohyblivé součásti soustavy. Poté dojde ke stanovení rychlosti rotace ramene, která je stejná pro všechny součásti, které jsou nuceny rotovat okolo stejné nepohyblivé součásti. Poměr počtu zubů (R) řídících a řízených součástí na cestě mezi řídící a nepohyblivou součástí nelze aplikovat na výslednou rychlost rotace součásti, protože by nebylo možné získat nulovou výslednou rychlost rotace nepohyblivé součásti na základě znalosti nenulové rychlosti rotace řídící součásti a nenulovém poměru R. Rychlost rotace ramene okolo rotační osy nepohyblivé součásti a relativní rychlost rotace nepohyblivé součásti vůči rotaci ramene jsou

gear	$\omega_{gear}[rad \cdot s^{-1}]$	$\omega_{arm}[rad \cdot s^{-1}]$	$\omega_{gear/arm}[rad \cdot s^{-1}]$
$P1$	ω_{P1}	ω_{arm}	$-\omega_{P3/arm} \cdot N_{P3}/N_{P1}$
$P2$	0	ω_{arm}	$-\omega_{arm}$
$P3$	ω_{P3}	ω_{arm}	$\omega_{P3} - \omega_{arm}$

Tabulka 2.1: Úhlové rychlosti součástí, které tvoří soustavu, která je zobrazena na obrázku 2.34, pro řídící součást $P3$ a nepohyblivou součást $P2$.

v absolutní hodnotě stejně velké, pouze mají opačné znaménko, aby výsledná rychlost rotace nepohyblivé součásti byla nulová (viz vzorec 2.15 pro soustavu znázorněnou na obrázku 2.34). Ze znalosti výsledné rychlosti rotace řídící součásti a rychlosti rotace ramene okolo nepohyblivé součásti se stanoví relativní rychlost rotace řídící součásti vůči rotaci okolo ramene. Na základě znalosti této hodnoty a poměru počtu zubů mezi příslušnou dvojicí součástí dojde k vypočtení všech relativních rychlostí rotace součástí vůči rotaci okolo ramene.

Pro soustavu, která je zobrazena na obrázku 2.34, jsou uvedeny v tabulce 2.1 stanovené výrazy, které jsou odvozeny ve vzorci 2.15 a 2.19, pro vypočtení úhlových rychlostí součástí dané soustavy ze znalosti rychlosti řídící součásti. Úhlová rychlost rotace ramene se pak vypočte dle vzorce 2.19. Tato hodnota se doplní do všech řádků tabulky pro sloupec rotace ramene (viz ω_{arm} v tabulce 2.1). Poměr počtu zubů R mezi požadovaným párem součástí se aplikuje na relativní úhlovou rychlost řízené součásti vůči rotaci ramene ($\omega_{gear/arm}$).

$$\omega_{P3/arm} = \omega_{P3} - \omega_{arm} \tag{2.13}$$
$$-\omega_{P2/arm} = \omega_{P3/arm} \cdot N_{P3}/N_{P2} \tag{2.14}$$
$$\omega_{P2/arm} = -\omega_{arm} \tag{2.15}$$

$$\omega_{P3/arm} \cdot N_{P3}/N_{P2} = -\omega_{P2/arm} \tag{2.16}$$
$$(\omega_{P3} - \omega_{arm}) \cdot N_{P3}/N_{P2} = \omega_{arm} \tag{2.17}$$
$$\alpha = N_{P3}/N_{P2} \tag{2.18}$$
$$\omega_{arm} = \omega_{P3} \cdot \alpha/(1+\alpha) \tag{2.19}$$

Metoda založená na algebraickém výpočtu vychází ze znalosti rychlosti rotace první a poslední součásti na analyzované cestě, kde první součástí je řídící součást a poslední součástí je nepohyblivá součást. Stanoví se poměr počtu zubů R stejným způsobem, který byl popsán v sekci 2.4.1 této podkapitoly. Relativní úhlová rychlost první součásti na cestě vůči rotaci ramene je rovna $\omega_{F/arm} = \omega_F - \omega_{arm}$, kde ω_F je výsledná rychlost první součásti na analyzované cestě. Relativní úhlová rychlost poslední součásti na cestě vůči rotaci ramene je rovna $\omega_{L/arm} = \omega_L - \omega_{arm}$, kde ω_L je její výsledná rychlost. Poměr relativní úhlové rychlosti vůči rotaci ramene poslední součásti a první součásti na analyzované cestě je roven poměru počtu zubů R na této cestě mezi první a poslední součástí a lze vyjádřit následujícím vztahem $sign \cdot R = (\omega_L - \omega_{arm})/(\omega_F - \omega_{arm})$, kde $sign$ je znaménko poměru počtu zubů R na této cestě mezi první a poslední součástí. Směr rotace součásti, která je donucena

rotovat okolo rotační osy nepohyblivé součásti, je opačný nežli by byl směr rotace nepohyblivé součásti, kdyby se mohla otáčet okolo své osy rotace. Hodnota znaménka pro dvojici součástí, které spolu mají kontakt, je rovna +1, pokud směr rotace řízené součásti je stejný jako směr rotace řídící součásti. V opačném případě je hodnota znaménka pro příslušnou dvojici součástí rovna hodnotě −1. Znaménko *sign* je rovno výsledku dílčích součinů znamének dvojice součástí, které spolu mají kontakt (tvoří hranu grafu), na cestě mezi řídící a nepohyblivou součástí. Postup pro stanovení parametrů animace součástí soustavy je při znalosti úhlové rychlosti rotace ramene okolo rotační osy nepohyblivé součásti shodný jako při použití výpočtu pomocí tabulky.

Aby bylo možné použít algebraickou metodu, kterou prezentuje Milton [25], pro stanovení rychlosti rotace součásti okolo rotační osy nepohyblivé součásti, je potřeba znát na analyzované cestě buď rychlost rotace první a poslední součásti nebo rychlost rotace ramene a jedné součásti. Poměr relativní úhlové rychlosti vůči rotaci ramene poslední součásti a první součásti na analyzované cestě je roven poměru počtu zubů R na této cestě (viz rovnice 2.20). Pro příklad soustavy mechanických součástí, která je znázorněna na obrázku 2.34, kde je součást $P2$ nepohyblivou součástí a součást $P3$ je řídící součástí, se vypočte rychlost rotace ramene dle vzorce 2.23.

$$sign \cdot R = (\omega_L - \omega_{arm})/(\omega_F - \omega_{arm}) \tag{2.20}$$

$$sign \cdot N_F/N_L = (\omega_L - \omega_{arm})/(\omega_F - \omega_{arm}) \tag{2.21}$$

$$-N_{P3}/N_{P2} = (0 - \omega_{arm})/(\omega_{P3} - \omega_{arm}) \tag{2.22}$$

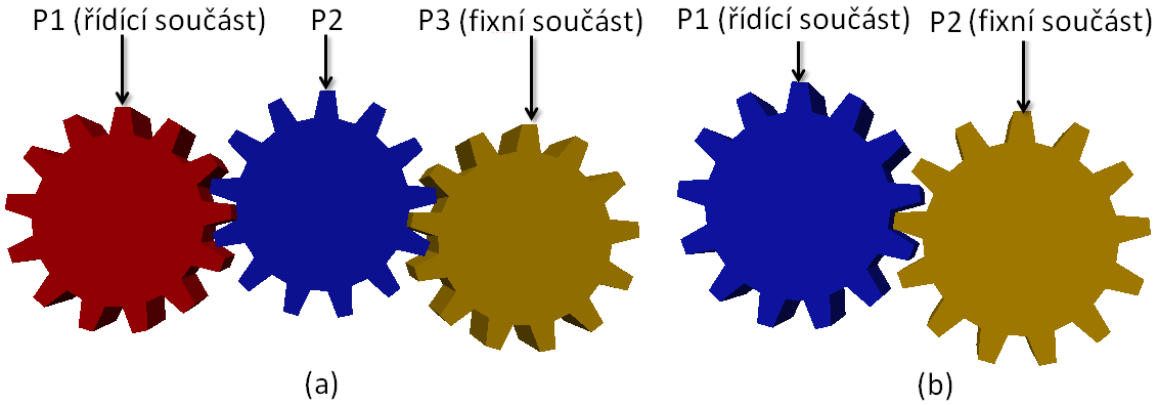
$$\omega_{arm} = \omega_{P3} \cdot N_{P3}/(N_{P3} + N_{P2}) \tag{2.23}$$

Obě dvě uvedené metody, které popisuje Milton [25] a které stanoví rychlost rotace součásti okolo rotační osy nepohyblivé součásti, mají stejné omezení, že je nelze použít pro případ, když je jednotkový poměr počtu zubů R na cestě mezi řídící a nepohyblivou součástí a zároveň když je hodnota znaménka rovna +1. Proč tomu tak je, si ukážeme na dvou jednoduchých soustavách součástí, které jsou zobrazeny na obrázku 2.35. Výrazy pro vypočtení rotačních ramen těchto dvou soustav budou uvedeny pro algebraickou metodu, ale i pro metodu výpočtu rychlosti rotace součásti okolo rotační osy nepohyblivé součásti s využitím tabulky je příčina omezení stejná, a proto příčina omezení těchto dvou metod bude ukázaná pouze na algebraické metodě. Na obrázku 2.35(a) je zobrazena soustava, pro kterou se projeví omezení obou dvou uvedených metod pro výpočet rychlosti rotace ramene okolo rotační osy nepohyblivé součásti. Hodnota jmenovatele výrazu 2.26 pro výpočet rotace ramene součástí soustavy, která je zobrazena na obrázku 2.35(a), při použití algebraické metody bude rovna nule. Dělením nulou vznikne nedefinovaná hodnota tohoto výrazu. Pro případ soustavy, která je zobrazena na obrázku 2.35(b), je sice poměr počtu zubů R na cestě mezi řídící a nepohyblivou součástí roven jedné, ale hodnota znaménka je rovna −1. Hodnota jmenovatele výrazu 2.29 pro výpočet rotace ramene okolo rotační osy nepohyblivé součásti je nenulová a velikost rychlosti rotace ramene je definovaná hodnota.

$$N_1/N_3 = (0 - \omega_{arm})/(\omega_1 - \omega_{arm}) \tag{2.24}$$

$$N_1 \cdot \omega_1 - N_1 \cdot \omega_{arm} = -N_3 \cdot \omega_{arm} \tag{2.25}$$

$$\omega_{arm} = \omega_1 \cdot N_1/(N_1 - N_3) \tag{2.26}$$



Obrázek 2.35: (a) Soustava, u které se projeví omezení použitých metod pro vypočtení rotace ramene okolo rotační osy fixní součásti. (b) Soustava, u které se neprojeví omezení použitých metod pro vypočtení rotace ramene.

$$-N_1/N_2 = (0 - \omega_{arm})/(\omega_1 - \omega_{arm}) \tag{2.27}$$

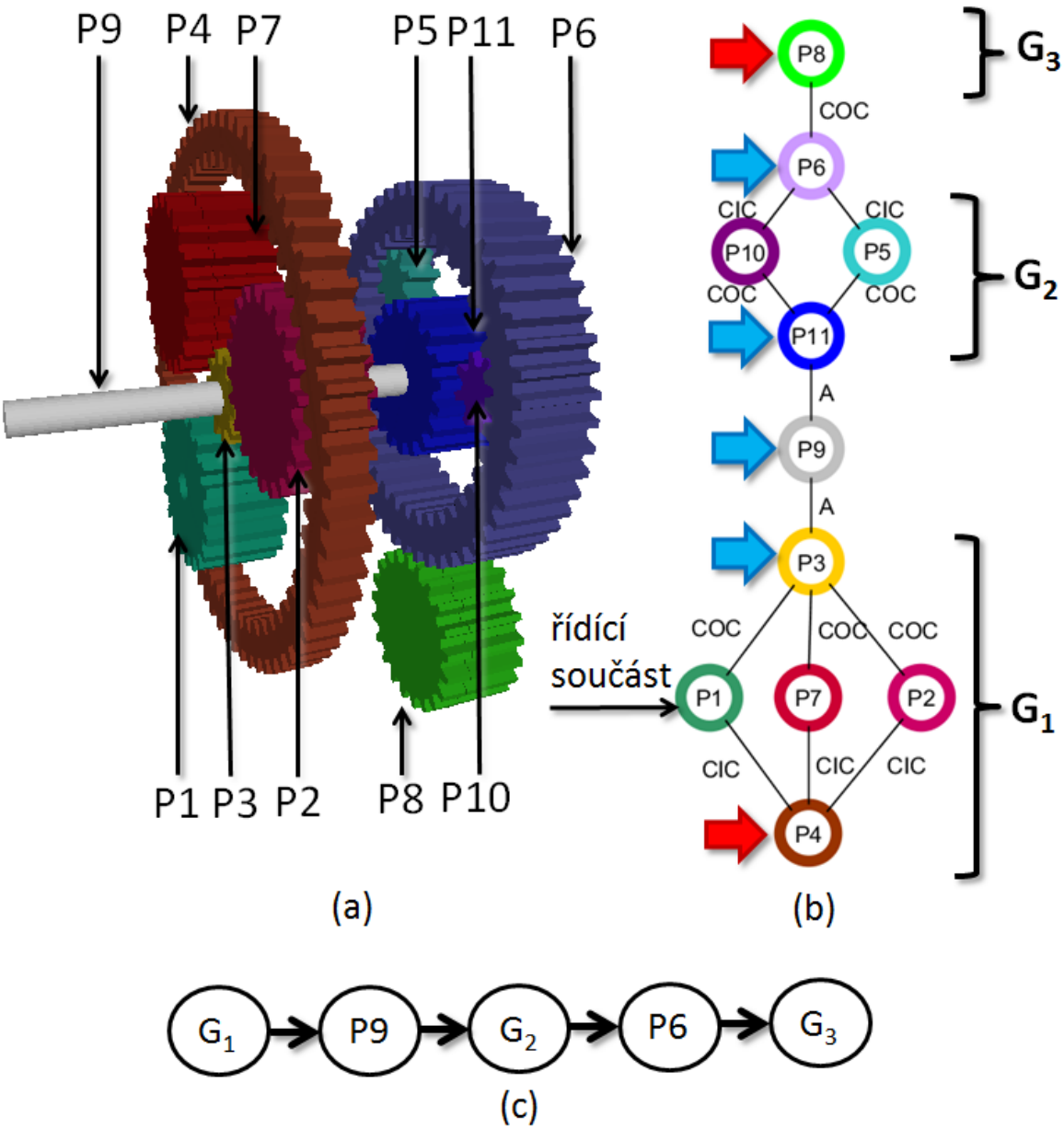
$$-N_1 \cdot \omega_1 + N_1 \cdot \omega_{arm} = -N_2 \cdot \omega_{arm} \tag{2.28}$$

$$\omega_{arm} = \omega_1 \cdot N_1/(N_1 + N_2) \tag{2.29}$$

2.4.6 Rozklad grafu interakce na podgrafy a stanovení, zda je animace soustavy proveditelná

Graf interakce se v této práci rozloží na podgrafy pro získání správných parametrů animace pohybu soustavy, aby došlo k nalezení správné cesty mezi řídící součástí a nepohyblivou součástí a aby došlo ke správnému složení rotace součásti okolo vlastní osy rotace a osy rotace nepohyblivé součásti, pokud jsou tyto osy koaxiální, do jednoho výsledného pohybu. Požadavek na sloučení uvedených pohybů je v této práci z důvodu, že pokud řídící součást je nucena rotovat okolo osy rotace nepohyblivé součásti, pak pokud nebude sloučen její pohyb okolo své osy rotace a osy rotace nepohyblivé součásti, pak bude nucena rotovat okolo stejné nepohyblivé součásti i řízená součást.

Princip stanovení parametrů animace součástí soustavy bude vysvětlen na soustavě mechanických součástí, která je zobrazena na obrázku 2.36(a). Na obrázku 2.36(b) označují červené šipky nepohyblivé součásti soustavy a modré šipky značí součást, která má osu rotace koaxiální s některou dostupnou nepohyblivou součástí soustavy. Dostupná součást soustavy představuje takovou součást, pro kterou existuje cesta mezi ní a řídící součásti, na které se nevyskytuje žádná nepohyblivá součást kromě ní, pokud by byla nepohyblivou součástí. Podgraf G grafu H je takový graf, který obsahuje podmnožinu vrcholů a podmnožinu hran grafu H. Pokud podgraf G neobsahuje vrchol V_i grafu H, pak ani v podgrafu G neexistují hrany grafu H, které vedou z/do vrcholu V_i v grafu H. Graf interakce soustavy (viz obrázek

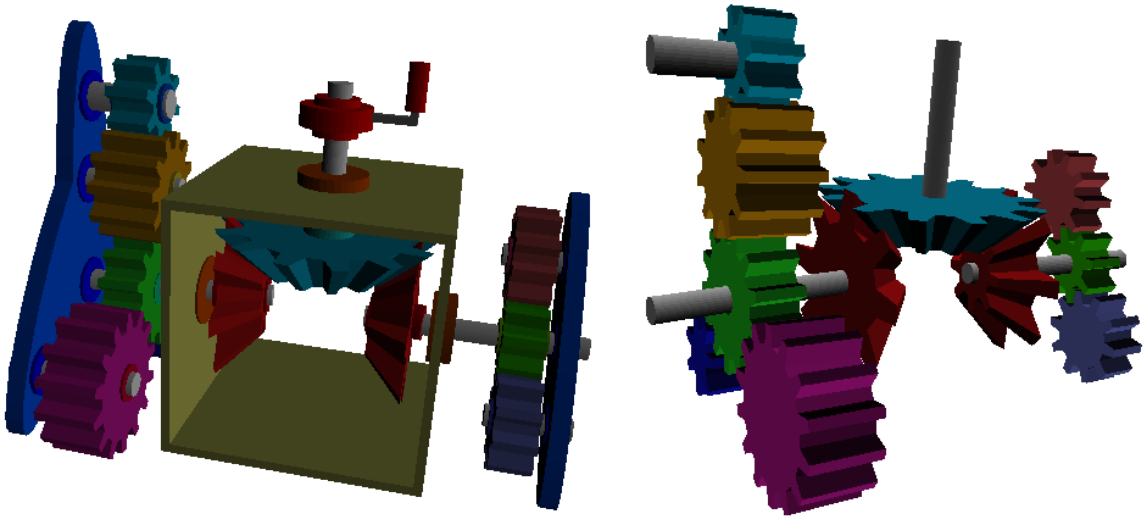


Obrázek 2.36: (a) Analyzovaná soustava. (b) Graf interakce a jeho rozklad na podgrafy. (c) Shlukování grafu interakce.

2.36(b)) se rozloží na konečnou množinu podgrafů, které jsou na obrázku označeny jako G_1, G_2 a G_3. Shlukování (z angl. clustering) seskupí množinu vrcholů a hran do jednoho uzlu, z/do něhož vedou takové hrany grafu interakce, které jsou spojeny s uzly, které nejsou obsaženy ve vytvořeném shluku. Pomocí procesu shlukování se vytvoří podgrafy grafu interakce. Uzly, které vzniknou při procesu shlukování jsou označeny na obrázku 2.36(c) jako uzel G_1, G_2 a G_3. Aby bylo možné rozdělit graf interakce na podgrafy, tak je potřeba nalézt všechny dostupné nepohyblivé součásti soustavy. Podgraf grafu interakce se vytvoří tak, že se prochází graf interakce do šířky směrem od řídící součásti a pokud je nalezen takový uzel V_i, který představuje součást, která má osu rotace koaxiální s osou rotace dostupné nepohyblivé součásti a nebo je nepohyblivou součástí, pak se průchod grafem v tomto směru ukončí pro aktuálně vytvářený shluk. Pokud nepředstavuje uzel V_i grafu interakce nepohyblivou součást, pak se tento uzel stane počátečním uzlem pro následující průchod grafem pro tvoření nového shluku a grafem interakce se prochází přes takové hrany, které ještě nebyly navštíveny ve fázi vytváření podgrafů grafu interakce. Takto vymezené části grafu interakce pak tvoří jeho podgrafy.

Typ analyzované součásti bude nastaven na statický, pokud bude mít tato součást kontakt se součástí soustavy, která byla donucena být statickou součástí, aby nedošlo ke změně množiny součástí, s nimiž má analyzovaná součást kontakt. Statický typ součásti bude nastaven i pro součást, jejíž osa rotace je koaxiální s osou rotace nepohyblivé součásti, aby opět nedošlo ke změně množiny součástí, s nimiž má analyzovaná součást kontakt. Nyní si vysvětlíme na příkladě soustavy, která je zobrazena na obrázku 2.36(a), k čemu by došlo, kdyby nebylo stanoveno toto chování. Všechny pohyblivé součásti soustavy představují dynamické typy součástí. Nejprve jsou vypočteny rychlosti rotace součástí, které tvoří podgraf G_1, protože je obsažena řídící součást v tomto podgrafu. Protože podgraf G_1 obsahuje nepohyblivou součást $P4$, tak je vypočtena rotace součástí v tomto podgrafu okolo osy rotace nepohyblivé součásti $P4$. Protože je osa rotace součásti $P3$ koaxiální s osou rotace nepohyblivé součásti, dojde ke sloučení rotace této součásti okolo své rotační osy a osy rotace nepohyblivé součásti. Dojde k vypočtení rychlosti rotace hřídele a součástí, které tvoří podgraf G_2. Poté dojde k vypočtení rychlosti rotace součásti $P6$ a k analyzování součástí podgrafu G_3, který tvoří nepohyblivá součást $P8$. Pokud by nebyl přenastaven typ součásti $P6$ z dynamické součásti na statickou součást, pak by došlo k tomu, že by byla nastavena rotace součásti $P6$ okolo osy rotace nepohyblivé součásti $P8$. Kolize parametrů animace součásti $P6$ by nenastala, protože by se tato součást jevila tak, že může rotovat okolo libovolné nepohyblivé součásti, ale tomu tak není, protože pokud by součást $P6$ začala rotovat okolo rotační osy součásti $P8$ a součásti $P5$, $P10$, $P11$ a $P9$ by rotovaly okolo vlastní rotační osy, pak by došlo k tomu, že by se vyjmenované součásti, které jsou uvnitř součásti $P6$, ocitly vně součásti $P6$ v průběhu animace soustavy. Toto chování není žádoucí, a proto je potřeba nastavit typ součásti $P6$ na statickou součást, aby došlo k vyhodnocení situace, že animace soustavy není proveditelná.

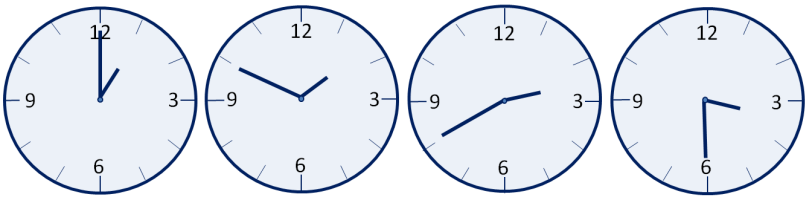
Pořadí zpracování jednotlivých podgrafů při výpočtu parametrů animace součástí soustavy je shodné s pořadím jejich vytváření. Pro každý podgraf se vypočítají parametry animace součástí, které tvoří daný podgraf. Po zpracování všech vytvořených podgrafů dojde k nastavení parametrů animace pomocných součástí soustavy, pokud takové součásti v soustavě existují. Pomocná součást soustavy je taková součást, pro kterou se nepočítá detekce kolizi s objekty ve scéně, nestanovují se její vlastnosti a ani se pro ní nevytváří šipky. Pro



Obrázek 2.37: (vlevo) Soustava s pomocnými součástmi. (vpravo) Soustava bez pomocných součástí.

tyto součásti je pouze stanoveno, zda se mají pohybovat a pokud ano, pak s jakou součástí soustavy, která není pomocnou součástí, mají mít totožný pohyb. Pomocné součásti soustavy slouží jako kontext pro pochopení funkce zobrazené soustavy součástí (viz obrázek 2.37). Podpora pomocných součástí soustavy v této práci byla vytvořena pro lepší pochopení funkce zobrazené soustavy a proto, že tímto způsobem lze animovat i pohyb součástí, které vykonávají stejný pohyb jako některá součást soustavy a které mají komplikovanější tvary nežli válec či kužel. Pomocné součásti dávají pozorovateli přídavnou informaci o tom, jaké prostorové uspořádání mají součásti vůči sobě v soustavě a jak jsou součásti v prostoru upevněny. Pomocné součásti nejsou z důvodu přehlednosti v této práci vykresleny do vytvořené statické ilustrace.

Pokud v analyzované soustavě mechanických součástí existuje alespoň jedna nepohyblivá součást a existuje více cest v analyzovaném podgrafu grafu interakce této soustavy mezi řídící součástí a nepohyblivou součástí, které mají odlišný poměr počtu zubů, pak není pohyb takovéto soustavy proveditelný. Protože je hřídel neozubenou součástí, pak pokud je hřídel zvolena za řídící součást soustavy a dochází ke stanovení rychlosti rotace řídící součásti okolo osy rotace nepohyblivé součásti, pak se do poměru počtu zubů pro řídící součást vezme počet zubů takové součásti, která má kontakt s řídící součástí a která je zároveň na analyzované cestě mezi řídící a nepohyblivou součástí soustavy. Pokud má součást nastaveny parametry pohybu, pak je potřeba stanovit, zda nekolidují nově vypočtené parametry s jejími starými hodnotami (směr rotace, velikost rychlosti rotace a osa rotace). Dále je potřeba zkontrolovat, zda je s definovanými typy součástí proveditelná animace analyzované soustavy.



Obrázek 2.38: Aliasing v časové oblasti při snímání animace hodin z výchozí polohy (první snímek) po 50 minutách ve 4 po sobě následujících snímcích. Při pozorování pohybu hodin hodinová ručička rotuje směrem doleva, kdežto minutová ručička rotuje směrem doprava.

2.4.7 Aliasing dynamické vizualizace soustavy mechanických součástí

Během animace soustavy mechanických součástí může dojít při zvolení „nevhodné“ úhlové rychlosti řídící součásti k tzv. aliasingu[36] v časové oblasti, který se projeví tak, že se pozorovateli zdá, že ozubené kolo rotuje opačným směrem, nežli by očekával, nebo zůstává statické. Aliasing vzniká v případě, kdy se vzorkuje spojitá informace nevhodnou vzorkovací frekvencí, při níž dochází k chybné interpretaci zobrazené informace z navzorkovaných hodnot. Vzorkování je proces, při němž se v definovaných časových okamžicích zjišťuje, jakou funkční hodnotu má analyzovaná spojitá informace v daný čas. Během procesu vzorkování vzniká chyba vzorkování, protože je vyjádřena spojitá informace konečným počtem diskrétních hodnot. Rotace ozubeného kola je principiálně shodná s animací hodin, na kterých si ukážeme, kdy u nich bude docházet k aliasingu. Na obrázku 2.38 jsou zobrazeny 4 pořízené snímky hodin po 50 minutách, kde první snímek (2.38 vlevo) odpovídá výchozí pozici ručiček hodin a každý další snímek je pořízen po uplynutí 50 minut od předchozího pořízeného snímku. Při animaci hodin dochází k tomu, že se minutová ručička otočí každou minutu o jeden stupeň. Pokud například se pořizuje snímek hodin každých 50 minut, pak se zdá, že minutová ručička rotuje směrem doprava, i když ve skutečnosti rotuje doleva. Tato skutečnost se děje z důvodu, že není splněn Shannonův vzorkovací teorém[36]. Pro vysvětlení Shannonova teorému je potřeba znát pojem vzorkovací frekvence. Vzorkovací frekvence stanovuje počet vzorků načtených za jednotku času při vzorkování spojité informace. Shannonův teorém říká, že vzorkovací frekvence f_s musí být nejméně dvakrát větší nežli dvojnásobek maximální frekvence f_{max} obsažené ve vzorkované informaci. Pokud by vzorkovací frekvence byla rovna dvojnásobku maximální frekvence, pak by byly pořízeny 2 snímky hodin po 30 minutách. Hodnota f_s $[s^{-1}]$ je rovna $1/T_s$, kde T_s je vzorkovací perioda $[s]$, která udává čas mezi pořízenými vzorky. Ukažme si, že pokud je pořízen snímek animace hodin v okamžiku, kdy minutová ručička bude oproti své výchozí poloze otočena o 180 stupňů, pak nedojde k aliasingu, protože bude splněna níže uvedená nerovnost pro Shannonův teorém. Z uvedeného vzorce pro teorém vyplývá, že pokud se uvažuje úloha animace hodin, pak je potřeba, aby se pořídily minimálně 2 snímky během uplynutí jedné hodiny od výchozího času, aby byla animace hodin bez aliasingu.

$$
\begin{array}{ll}
T_s & = 30 \text{ minut} = 1800 \text{ sekund} \\
T_{max} & = 60 \text{ minut} = 3600 \text{ sekund} \\
f_s & \geq 2 \cdot f_{max} \\
1/T_s & \geq 2 \cdot (1/T_{max}) \\
1/1800 & \geq 2 \cdot (1/3600) \\
1/30 & \geq 1/30
\end{array}
$$

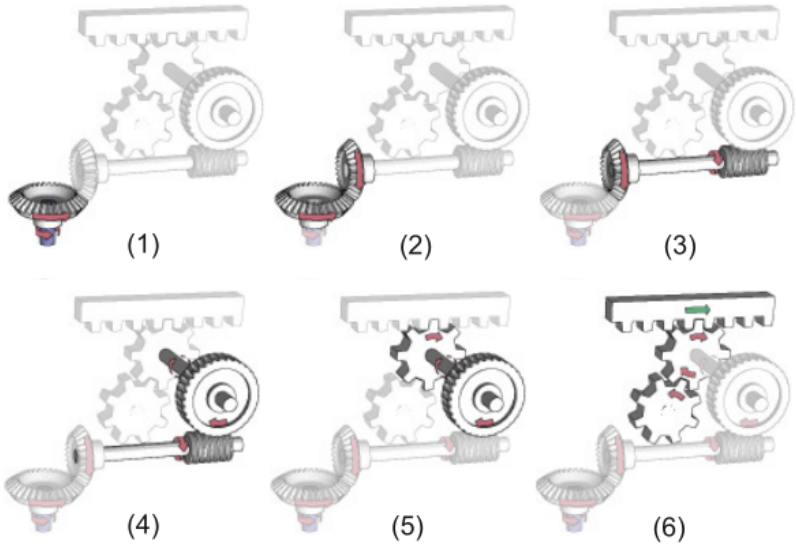
Pokud však zjišťujeme pozici minutové ručičky po uplynutí 50 minut oproti její předchozí poloze, pak dojde k aliasingu, protože nebude splněn Shannonův teorém z důvodu, že se zvětšuje vzdálenost mezi vzorky, a tím pádem se snižuje počet pořízených vzorků za jednotku času (vzorkovací frekvenci). Proto se bude uživateli zdát, že minutová ručička rotuje proti očekávanému směru rotace. Tento jev je označován jako stroboskopický efekt, který je optickým klamem. Konkrétněji se jedná o tzv. „wagon-wheel effect“, při jehož výskytu pozorovateli připadá, že se rotující objekty (např. hodiny, ozubená kola) otáčejí opačným směrem, nežli je předpokládaný směr jejich rotace, či jsou statické. Tohoto efektu si můžeme například všimnout ve westernových filmech, kdy je použita filmová kamera, která má frekvenci vzorkování 24 nebo 25 obrázků za sekundu. Frekvence vzorkování kamery je odvozena na základě fyziologických vlastností lidského oka (především setrvačnosti), pomocí níž vnímá pozorovatel posloupnosti obrázků, které jsou promítány určitou rychlostí sekvenčně, jako spojitý pohyb.

2.4.8 Omezení dynamické vizualizace soustavy mechanických součástí

V této práci není povoleno, aby řídící součástí byla pomocná součást. V této práci je podporována možnost vybrat pouze jednu součást soustavy jako řídící součást. Pokud soustava obsahuje nepohyblivou součást, znaménko rotace ramene je rovno +1 a poměr počtu zubů na analyzované cestě mezi řídící součástí a nepohyblivou součástí je roven 1, pak není možná animace pohybu soustavy při zvolené řídící součásti z důvodu omezení algebraické metody pro výpočet rotace ramene.

2.5 Vytvoření statické ilustrace soustavy součástí s grafickými informacemi

Pokud by měl pozorovatel k dispozici pouze statickou ilustraci složitější soustavy součástí bez jakékoliv pomocné grafické a textové informace, která by popisovala funkci dané soustavy, pak by porozumění takovéto ilustraci nebylo pro pozorovatele postačující. Grafické informace ve statické ilustraci soustavy mechanických součástí slouží jako přídavná informace, která má pomoci uživateli lépe pochopit, jak vyobrazená soustava funguje. Grafickými informacemi mohou být například šipky, klíčové snímky soustavy či kauzální řetězec. Pro všechny zmíněné techniky platí, že pro ně dochází k pořízení jednoho či několika snímků zobrazené soustavy z takové pozice, na které bude nejvíce zřetelná vysvětlovaná funkce. Zvolení pohledu na soustavu, ze kterého je pořízen snímek, má zásadní vliv jak na pochopení funkce soustavy tak na představu, jaké prostorové uspořádání mají součásti v soustavě. Problematika vytvoření pomocných grafických informací pro statickou ilustraci soustavy je ztížena skutečností, že



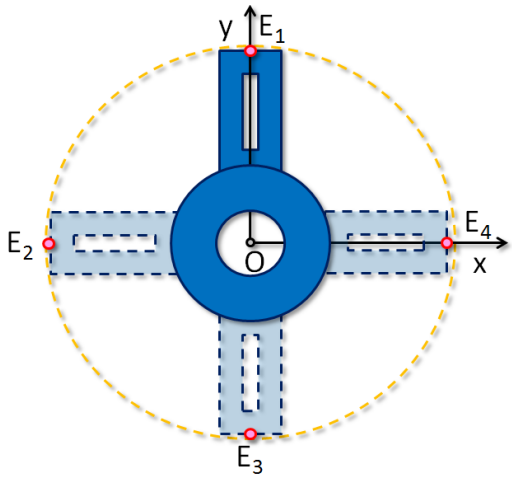
Obrázek 2.39: Kauzální řetězec soustavy mechanických součástí. Zdroj obrázku [27].

se nejedná o optimalizační problém, ve kterém by bylo jednoznačně definované optimum. Pro tuto úlohu neexistuje jasně definované optimální řešení, protože se jedná o subjektivní estetický vizuální vjem.

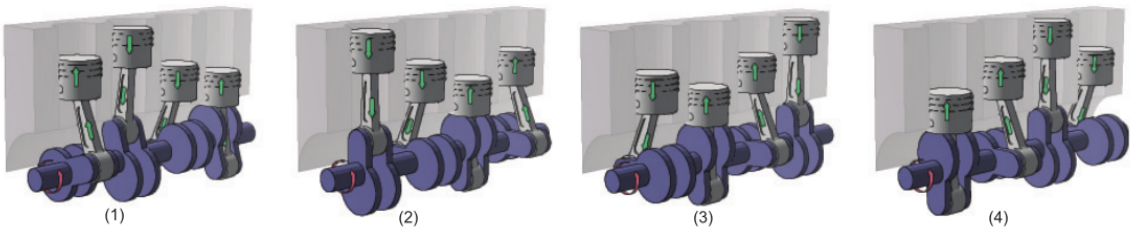
V následujícím textu bude uveden nejprve princip techniky pomocné grafické informace využívající kauzálního řetězce. Poté bude uveden princip techniky, který vytvoří klíčové snímky soustavy mechanických součástí. V neposlední řadě bude uvedena technika, která je realizována v této práci a která se zabývá umísťováním šipek do statické ilustrace analyzované soustavy mechanických součástí.

2.5.1 Kauzální řetězec

Mitra a kol. [27] prezentují techniku pro vytvoření kauzálního řetězce soustavy součástí, která je založena na zvýraznění šíření kinetické energie soustavou směrem od řídící součásti. Nechť hrana grafu interakce existuje mezi uzly P_i, P_j a kinetická energie se šíří směrem od součásti P_i, pak řízenou součástí je součást P_j a řídící součástí je P_i. V jednotlivých časových okamžicích se zvýrazní na snímku soustavy součástí takové součásti, které jsou buď aktuálně zpracovávanou řídící součástí a nebo řízenou součástí. Účelem zvýraznění součástí v příslušném časovém okamžiku je, že tyto součásti mají poutat pozornost uživatele a ostatní součásti, jejichž barevnost je potlačena, a které mohou být částečně zprůhledněny, slouží pouze jako kontext. Pro aktuálně zvýrazněné součásti jsou vytvořeny šipky, které znázorňují pohyb těchto součástí. Příklad kauzálního řetězce pro soustavu mechanických součástí je zobrazen na obrázku 2.39.



Obrázek 2.40: Znázornění součásti v jejích krajních polohách pro techniku klíčových snímků.



Obrázek 2.41: Klíčové snímky soustavy mechanických součástí, která je tvořena písty. Zdroj obrázku [27].

2.5.2 Klíčové snímky

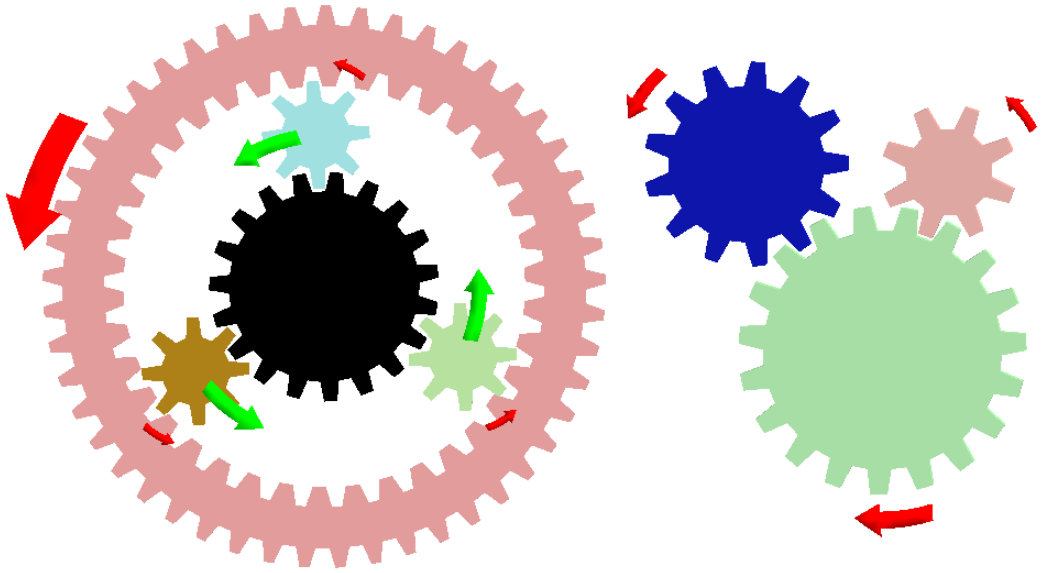
Pro soustavu součástí, která vykonává periodický pohyb, je vhodné využít techniku klíčových snímků, aby pozorovatel lépe porozuměl, jak funguje zobrazená soustava mechanických součástí. Mitra a kol. [27] prezentují techniku pro vytvoření klíčových snímků soustavy součástí. Statická ilustrace v tomto případě obsahuje několik klíčových snímků, které byly pořízeny v takových kritických časových okamžicích, které jsou zásadní pro pochopení, jaký pohyb vykonávají součásti uvnitř soustavy. Klíčové snímky musí být zobrazeny v chronologickém pořadí, aby mohl pozorovatel mentálně zrekonstruovat pomocí interpolace polohy součásti ze dvou sousedních snímků, jak se změnila poloha součásti mezi těmito snímky. Může se jednat o takové snímky, v nichž jsou jednotlivé součásti zobrazeny v krajních polohách svého pohybu.

Pojem krajní poloha pohybu součásti bude vysvětlen na obrázku 2.40. Extrémní bod součásti, který je označen na obrázku jako E_1, je takový bod součásti, který má v rovině xy maximální souřadnici y z množiny bodů, které tvoří povrch znázorněné součásti. Na obrázku je znázorněna iniciální pozice součásti jako modrý objekt. Žlutá kružnice znázorňuje pozici tohoto extrémního bodu během otáčení objektu okolo jeho rotační osy. Krajní poloha součásti je taková její poloha, ve které má extrémní bod maximální či minimální hodnotu souřadnice x nebo y na zobrazené žluté kružnici. Na znázorněné žluté kružnici jsou tyto extrémní body označeny jako E_1, E_2, E_3 a E_4. Čárkované hranice objektu znázorňují pozici objektu v jeho krajních polohách po rotaci objektu okolo jeho rotační osy, která prochází bodem O.

Dalo by se říci, že tato technika je vhodná na zobrazení soustav, jejichž pohyb není pro uživatele na první pohled zcela zřejmý. Použití této techniky například pro soustavu ozubených kol, pro které se množina kolidujících součástí během animace nemění, by nedával příliš smysl, protože by tato technika nedala pozorovateli skoro žádnou přídavnou informaci oproti použití techniky šipek. Na obrázku 2.41 je zobrazena soustava pístů, na kterou byla použita technika ilustrace pohybu za použití klíčových snímků. Předpovědět pohyb zobrazené soustavy pístů není již tak jednoduchá záležitost, protože se jedná o komplexnější pohyb. Na časových snímcích jsou zobrazeny součásti, které vykonávají periodický pohyb, v jejich krajních polohách. Pozorovatel mentálně zrekonstruuje, jak bude vypadat spojitý pohyb těchto součástí. V každém snímku jsou zároveň zobrazeny šipky, které znázorňují, jak se součásti v daný okamžik pohybují.

2.5.3 Šipky

Přidání šipek do vytvořeného snímku soustavy součástí není triviálním problémem, protože při stanovení umístění šipky je potřeba uvažovat nejen, aby byla informace zobrazená na obrázku zřetelná, ale i aby nedošlo k takovému zakrytí objektů ve scéně, při kterém by byla ztracena představa o prostorovém uspořádání součástí v soustavě. Pro případ umístění šipky ve 3D je potřeba stanovit, jak bude šipka v prostoru natočena (3 DOF), umístěna (3 DOF), jakou bude mít tloušťku (1 DOF), délku (1 DOF) a výšku (1 DOF). Uvedené pojmy rozměrů šipky jsou znázorněny na obrázku 2.43. Šipka, která je znázorněna na obrázku 2.43(a), se umísťuje na podstavu objektu a šipka, která je znázorněna na obrázku 2.43(b), se umísťuje na plášť objektu. Stanovení, na jakou část objektu se umístí jaký typ šipky, bylo provedeno na základě požadavku, aby byla šipka co nejlépe vidět. Zkratka DOF (z angl. degrees of freedom) znamená počet stupňů volnosti systému, který určuje počet nezávislých parametrů

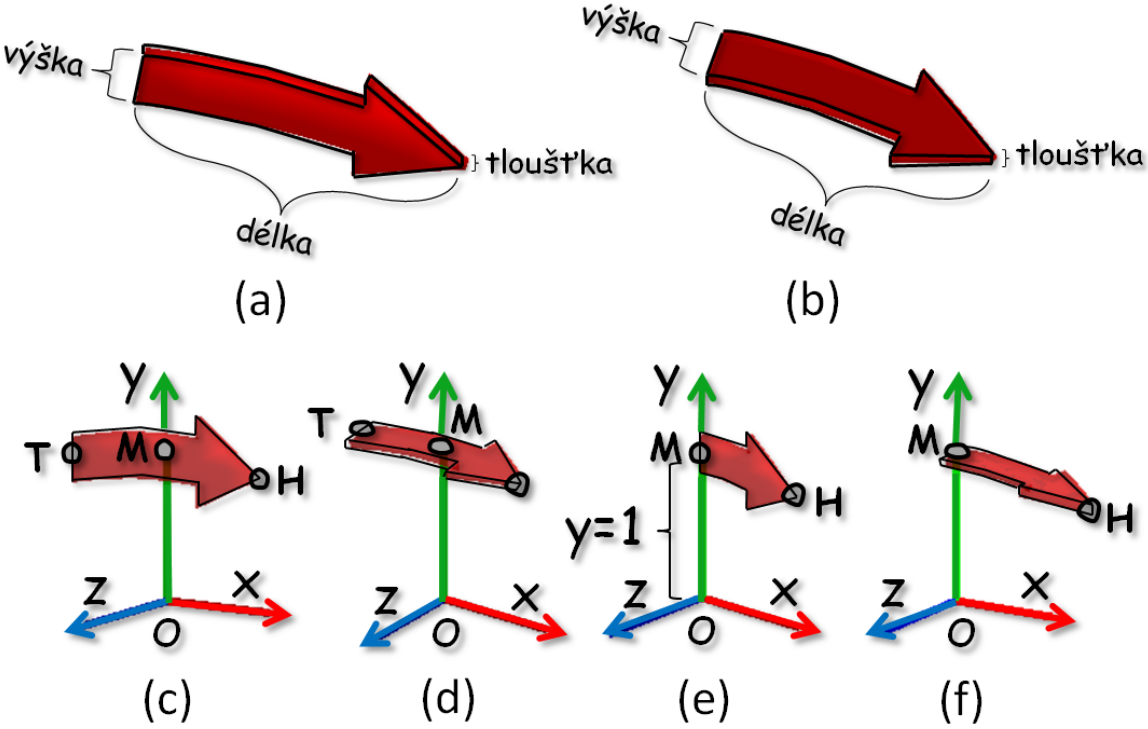


Obrázek 2.42: Statická ilustrace se šipkami pro soustavu součástí, (vlevo) ve které existuje nepohyblivá součást (černá barva) a (vpravo) která neobsahuje nepohyblivou součást.

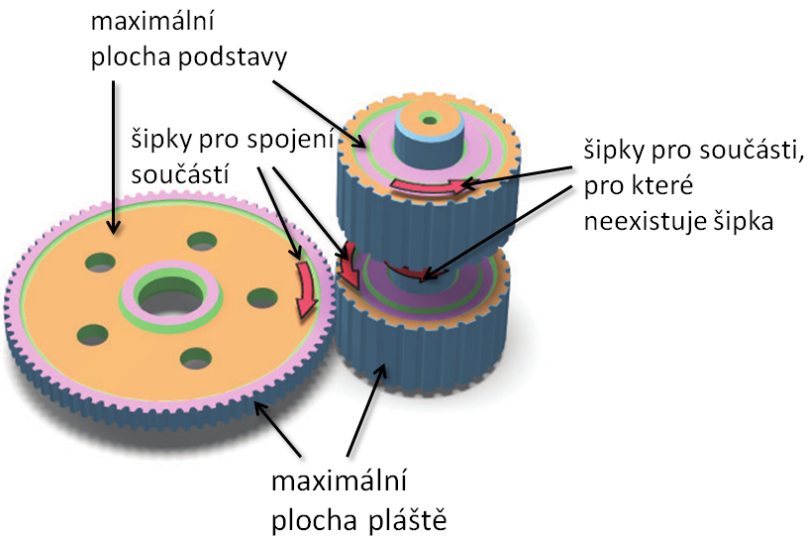
systému. Stupně volnosti pro natočení a umístění šipky ve 3D prostoru představují rotaci či posun ve směru souřadnicové osy x, y nebo z.

Příklad statické ilustrace pohybu soustavy součástí, která obsahuje nepohyblivou součást (černá součást), je znázorněna na obrázku 2.42 vlevo. Ostatní součásti soustavy jsou pohyblivé dynamické součásti. Rotace dynamické součásti soustavy okolo osy rotace nepohyblivé součásti je znázorněna zelenou šipkou. Rotace součásti okolo vlastní osy rotace je znázorněna červenou šipkou. Pro součást s vnitřními a vnějšími zuby, která je na obrázku 2.42 vlevo znázorněna růžovou barvou, není vytvořena šipka, která by znázorňovala rotaci této součásti okolo rotační osy nepohyblivé součásti, protože rotační osa této dynamické součásti je koaxiální s rotační osou nepohyblivé součásti, a proto je pro tuto součást vytvořena pouze červená šipka. Na obrázku 2.42 vpravo je znázorněn příklad soustavy mechanických součástí, která obsahuje pouze pohyblivé součásti.

Tato kapitola je členěna tak, že nejprve bude vysvětlen princip techniky statické ilustrace se šipkami vytvořené také pro spojení součástí, kterou prezentovali Mitra a kol. [27], a poté bude vysvětlen princip realizované metody v této práci. Pořadí dále uvedených metod je zvoleno z důvodu, že metoda umístění šipek, která je realizovaná v této práci, má některé společné rysy s technikou, kterou prezentovali Mitra a kol. [27]. Společnými rysy dále popisovaných dvou metod pro stanovení umístění šipek statické ilustrace, která znázorňuje pohyb součástí soustavy, je, že jsou podporovány stejné typy šipek pro rotační pohyb součásti a že je umožněno optimalizovat umístění šipek podle vzdálenosti od osy rotace součásti a podle vzdálenosti od těžiště součásti ve směru její osy rotace.



Obrázek 2.43: Znázornění pojmů pro rozměry šipky, která se umísťuje na (a) podstavu objektu nebo (b) plášť objektu.



Obrázek 2.44: Zobrazeny šipky vygenerované pro hranu grafu interakce a šipky pro součásti, pro které neexistuje žádná šipka. Texty obrázku byly přeloženy do češtiny autorkou práce. Zdroj obrázku [27].

Statická ilustrace se šipkami vytvořená také pro spojení součástí

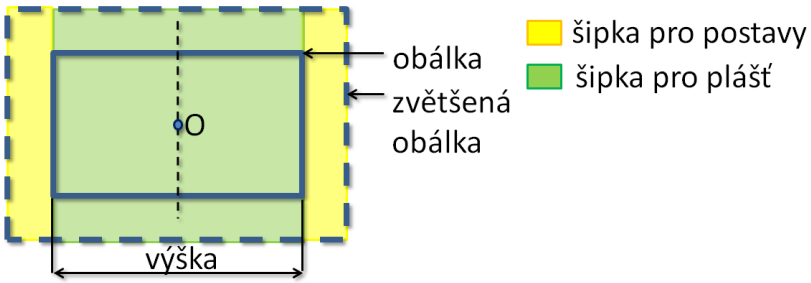
Mitra a kol. [27] prezentují techniku pro umístění šipek v soustavě součástí, která nejprve stanoví iniciální umístění šipek a poté provede optimalizaci umístění šipek tak, aby byly šipky co nejméně zakryty ostatními objekty ve scéně. Na obrázku 2.44 je znázorněn princip umísťování šipek do soustavy mechanických součástí, který prezentuje Mitra a kol. [27] a který bude nyní vysvětlen. Technika pro umístění šipek do soustavy součástí vychází ze znalosti grafu interakce pro soustavu součástí, který obsahuje informaci o párech součástí, které spolu mají kontakt, a typu jejich spojení. Pro každé spojení dvojice součástí kromě koaxiálního spojení se stanoví umístění šipky pro obě dvě součásti, které spolu mají kontakt. Je snaha umístit šipky, které byly vytvořeny pro hranu grafu interakce, co nejblíže k průsečíku těchto dvou součástí. Pro koaxiální spojení není vhodné vytvořit šipku, protože by došlo ke vzájemnému překryvu těchto šipek z důvodu, že hřídel je uvnitř ozubené součásti a směr obou dvou součástí je stejný. Rotační pohyb součásti je označen buď šipkou, která je umístěna na plášti součásti (viz obrázek 2.43(b)), a nebo šipkou, která je umístěna na podstavě objektu (viz obrázek 2.43(a)). Podle typu spojení se stanoví podmínky pro umístění šipek, které jsou vytvořeny pro spojení dvojice součástí, které spolu mají kontakt. Tyto podmínky zajistí, že bude nalezeno takové umístění šipek, které bude pokud možno co nejpřehledněji znázorňovat pohyb součástí. Pokud je spojení dvojice součástí například typu válec na válci, pak se umístí šipky na podstavy obou dvou součástí. Po stanovení umístění šipek pro spojení dvojice součástí dojde k vygenerování šipek pro pohyblivé součásti, pro které doposud neexistuje žádná šipka. Počet šipek znázorňující pohyb součásti je závislý na rozměrech součásti. Pokud je součást například podlouhlý válec, pak je vhodné vygenerovat takové množství šipek umístěných na plášti objektu, které je úměrné délce tohoto válce. Mitra a kol. [27] využívají pro vytvoření statické ilustrace soustavy paměť hloubky (z–buffer) a techniku dotazů na zastínění (z angl. occlusion queries). Technika dotazů na zastínění objektu se používá pro stanovení míry viditelnosti objektu ve scéně, která se stanoví podle počtu viditelných fragmentů, které prošly testem hloubky. Pomocí testu hloubky se zjistí, že fragmenty objektu, které leží blíže ke kameře, budou zakrývat fragmenty objektu, které leží za nimi. Rozhodnutí, zdali bude ležet šipka znázorňující rotaci součásti na podstavě či plášti objektu, závisí na velikosti viditelné plochy pláště a podstav součásti. Je preferováno umístit šipku na takovou část (podstava či plášť) součásti, která má větší viditelnou plochu. Při hledání nejlepšího umístění šipky znázorňující pohyb součásti záleží na stanovené váze pro jednotlivá kritéria (např. viditelná velikost ploch částí součásti, vzdálenost šipek vygenerovaných pro spojení součástí). Šipky pro statickou ilustraci jsou vypočteny pro zvolenou pozici a natočení kamery. Mitra a kol. [27] umožnili pro nalezení lepšího umístění šipky znázorňující pohyb součásti změnit pro šipku, která je umístěna na plášti objektu (viz obrázek 2.43(b)), její vzdálenost od roviny, ve které leží počátek lokálního souřadnicového systému součásti (těžiště), a jejíž normálový vektor je roven směrovému vektoru osy rotace součásti. Dojde tak k posunu pozice šipky podél rotační osy součásti. Mitra a kol. [27] také umožnili změnit vzdálenost šipky od osy rotace součásti pro šipku, která je umístěna na podstavě součásti (viz obrázek 2.43(a)). Tyto dvě uvedené optimalizace mají za úkol pomoci nalézt takové umístění šipky ve snímku, pro které je šipka pokud možno co nejméně zakryta objekty ve scéně. Dalším požadavkem na optimální umístění šipky je, aby byla vidět celá hlavička šipky, která poskytuje pozorovateli informaci o směru pohybu součásti. Velikost šipky závisí na velikosti součásti, jejíž rotační pohyb má šipka znázorňovat.

Realizovaná metoda

Nejprve bude vysvětlen princip nalezení pozice šipky, která znázorňuje rotaci součásti okolo vlastní osy rotace, a poté bude vysvětlen princip nalezení pozice šipky, která představuje rotaci dynamické součásti okolo osy rotace nepohyblivé součásti.

V této práci je realizovaná statická ilustrace se šipkami, které znázorňují pohyb pohyblivých součástí soustavy, pro uživatelem definovaný pohled. Nejprve budou seřazeny počátky lokálních souřadnicových systémů součástí směrem od kamery, aby byly vytvořeny dříve šipky pro součásti, které jsou blíže ke kameře. Protože šipky, které znázorňují pohyb součásti, budou umístěny vně objektu, tak je potřeba nalézt zvětšenou obálku součásti, ve které budou vygenerovány vzorky, do kterých se pak umístí bod M šipky (viz obrázek 2.43). Šipka pak bude zarovnána s lokálním souřadnicovým systémem součásti tak, že souřadnicová osa z šipky se ztotožní s rotační osou součásti. Tvar zvětšené obálky součásti je shodný s tvarem součásti. Pro všechny součásti soustavy je stejná vzdálenost, o kterou je zvětšena obálka. Vzorky se vygenerují ve zvětšené obálce pro součást ve smyslu Manhattanské metriky, tzn. vzorky jsou umístěny v mřížce tak, že kolmá vzdálenost vzorku je ke všem sousedním vzorkům stejná. Vzorky se generují ve zvětšené obálce ve směru souřadnicových os obálky, které jsou shodné se souřadnicovými osami součásti. Grafická informace ve formě šipek bude generována pouze pro takové součásti soustavy, pro které bude zjištěno pomocí techniky dotazů na zakrytí objektu, že počet jejích viditelných fragmentů ve scéně a relativní počet jejích viditelných fragmentů je větší nežli definovaná mez. Relativní počet fragmentů říká, z kolika procent je viditelný objekt při vykreslení všech objektů ve scéně. Dále je zjištěno použitím dotazů na zakrytí, jaké vzorky jsou uvnitř zvětšené obálky viditelné. Vzorky mohou být zakryty buď objekty ve scéně nebo již vygenerovanými šipkami pro součásti, které jsou blíže ke kameře.

V této práci jsou podporovány dva základní typy šipek, které jsou znázorněny na obrázku 2.43(a,b). Prvním typem šipky je šipka, která se umísťuje na plášť objektu (viz obrázek 2.43(b)), a druhým typem šipky je šipka, která se umísťuje na podstavu objektu (viz obrázek 2.43(a)). Pro viditelné vzorky součásti se vykreslí oba dva typy šipek. Vykreslení obou dvou typů podporovaných šipek pro každý vzorek je provedeno z důvodu, že nelze jednoznačně určit, pro jaký vzorek v obálce je vhodnější uvažovat konkrétní typ šipky, protože počet viditelných fragmentů šipky je závislý na umístění a natočení kamery. Byly však stanoveny preferované typy šipek podle pozice vzorku ve zvětšené obálce součásti podle principu, který je znázorněn na obrázku 2.45. Pokud je vzdálenost vzorku od roviny, jejíž normálový vektor je roven směrovému vektoru rotační osy součásti, a ve které leží počátek lokálního souřadnicového systému součásti (těžiště) O, větší nebo rovna poloviční hodnotě výšky součásti, pak je preferovaným typem šipky šipka, která se vykreslí na podstavu součásti (viz obrázek 2.43(a)). V opačném případě je preferovaným typem šipky taková šipka, která se vykreslí na plášť součásti (viz obrázek 2.43(b)). Pokud se liší počet viditelných fragmentů uvedených dvou typů šipek o více jak definovanou mez, pak se neupřednostňuje preferovaný typ šipky pro daný vzorek, ale vygenerovanou šipkou bude šipka s větším počtem viditelných fragmentů. Jinak je pro daný vzorek vygenerován preferovaný typ šipky. Ohodnocení vzorku udává míru vhodnosti umístění šipky do daného vzorku. Výsledné ohodnocení pro nalezení nejlepšího typu šipky a nejvhodnějšího vzorku uvnitř zvětšené obálky se řídí dle dále popsaného principu ohodnocení vzorku a viditelnosti šipky.

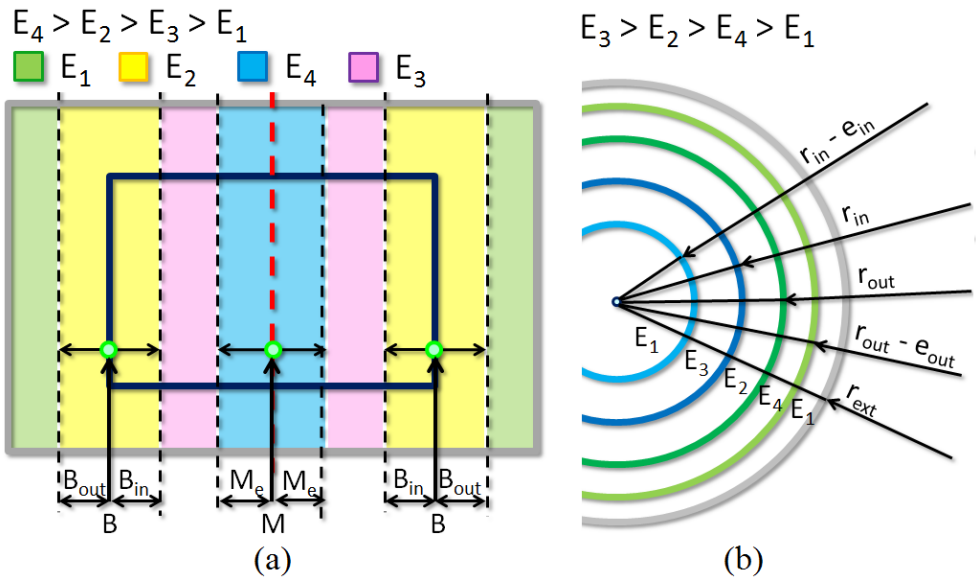


Obrázek 2.45: Preferované typy šipek podle pozice vzorku ve zvětšené obálce součásti.

Výsledné ohodnocení vhodnosti umístění šipky na pozici vzorku ve zvětšené obálce součásti je rovno $E = f_p \cdot E_p + f_a \cdot E_a$, kde f_p je faktor pozice, f_a je faktor počtu viditelných fragmentů šipky, E_a je ohodnocení šipky a E_p je ohodnocení pozice. Ohodnocení šipky odpovídá relativnímu počtu viditelných fragmentů šipky. Ohodnocení pozice je složeno z následujících kritérií $E_p = f_r \cdot E_r + f_s \cdot E_s + f_f \cdot E_f + f_w \cdot E_w$, kde f představuje faktor, E ohodnocení a významy dolních indexů jsou: r je poloměr, s je řez, f je síla (přitažlivá/odpudivá) a w je slábnutí. Hodnoty všech faktorů jsou v intervalu $< 0, 1 >$ a udávají váhu pro dané kritérium. Nyní budou podrobně vysvětlena uvedená jednotlivá kritéria. Podle estetického hlediska vhodnosti umístění šipky do konkrétního vzorku byly v této práci zvoleny hodnoty ohodnocení jednotlivých řezů v modelu řezů a slupek ve slupkovém modelu.

Velikost ohodnocení vzorku z hlediska poloměru (vzdálenosti vzorku od osy rotace) závisí na tom, do jaké slupky náleží daný vzorek. Hranice a šířky slupek slupkového modelu, který bude nyní popsán, jsou znázorněny na obrázku 2.46(b). Slupkový model je definován pomocí čtyř parametrů, kterými jsou r_{in} vnitřní poloměr, r_{out} vnější poloměr, e_{in} šířka slupky směrem od vnitřního poloměru ke středu slupkového modelu a e_{out} je šířka slupky směrem od vnějšího poloměru k hranici slupkového modelu, která je označena jako r_{ext}. Na obrázku je označeno ohodnocení jednotlivých slupek a velikosti uvedených čtyř ohodnocení jsou relativně vůči sobě porovnány v obrázku nahoře. Pro slupkový model platí, že hodnota vnějšího poloměru (r_{out}) je větší nebo rovna hodnotě vnitřního poloměru (r_{in}). Hodnoty všech čtyř uvedených proměnných, které definují slupkový model, jsou udány relativně k hodnotě poloměru zvětšené obálky součásti. Příklad ohodnocení viditelných vzorků součásti podle slupkového modelu je znázorněn na obrázku 2.47 nahoře.

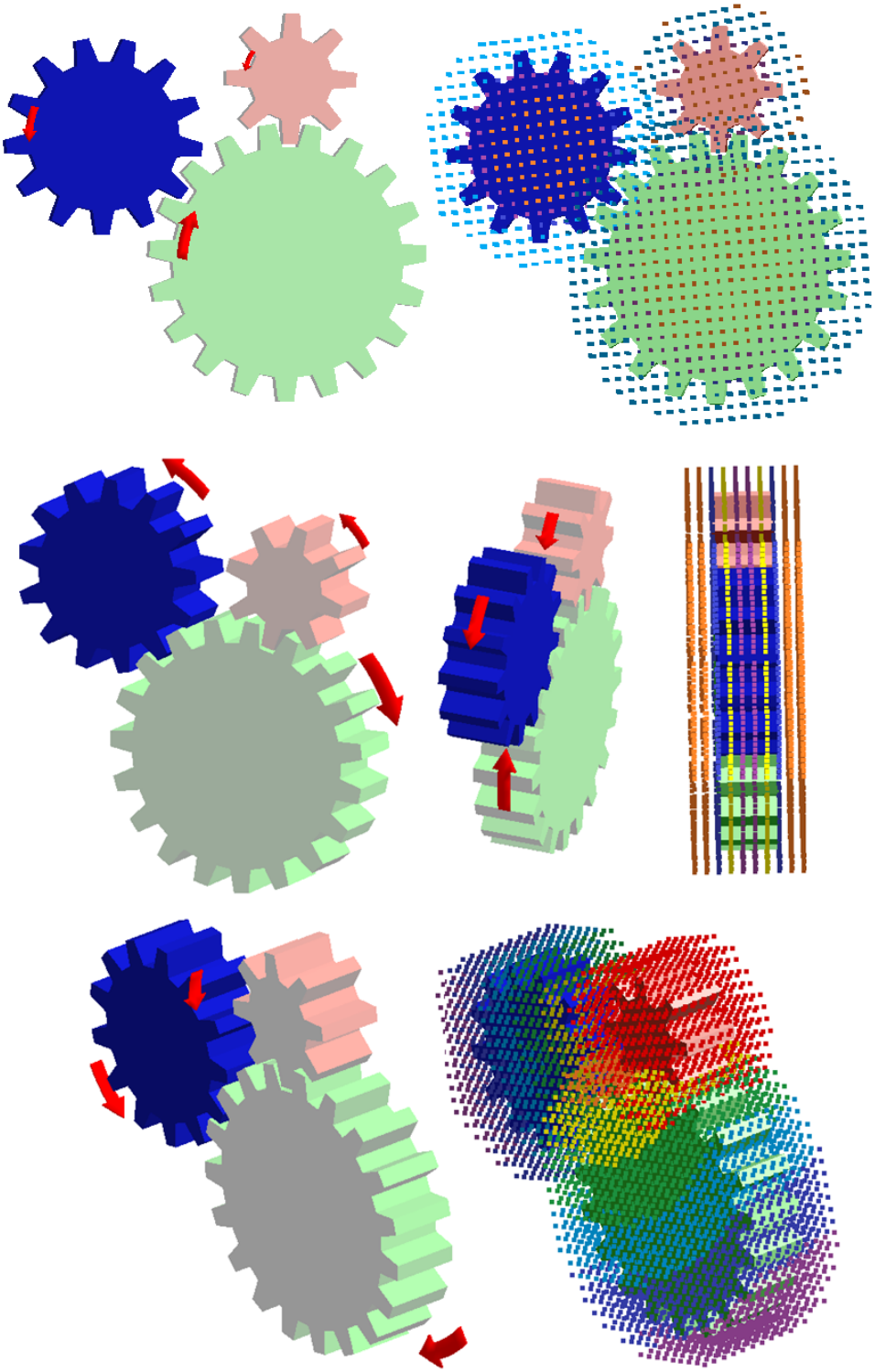
Rovina π je taková rovina, ve které leží lokální souřadnicový systém součásti (těžiště) a jejíž normálový vektor je roven směrovému vektoru rotační osy součásti. Ohodnocení vzorku podle řezu závisí na vzdálenosti vzorku od roviny π. Model řezů je znázorněn na obrázku 2.46(a) a je definován pomocí pěti parametrů, jimiž jsou bod B, bod M, šířka B_{in}, B_{out} a M_e. Řezy v modelu řezů jsou rovnoběžné s rovinou π. Bod M definuje vzdálenost prostřední řezné roviny od roviny π. Bod B definuje vzdálenost hraniční roviny od roviny π. Parametr M_e udává poloviční šířku prostředního řezu modelu řezů. Parametr B_{out} udává šířku řezu v modelu řezů roviny, ve které leží bod B, směrem k hranici zvětšené obálky součásti. Naopak parametr B_{in} udává šířku řezu v modelu řezů roviny, ve které leží bod B, směrem dovnitř obálky součásti k rovině π, která je znázorněna na obrázku čárkovanou červenou čarou. Jak je z obrázku patrné, tak šířka prostředního řezu je symetrická okolo roviny, v níž leží bod



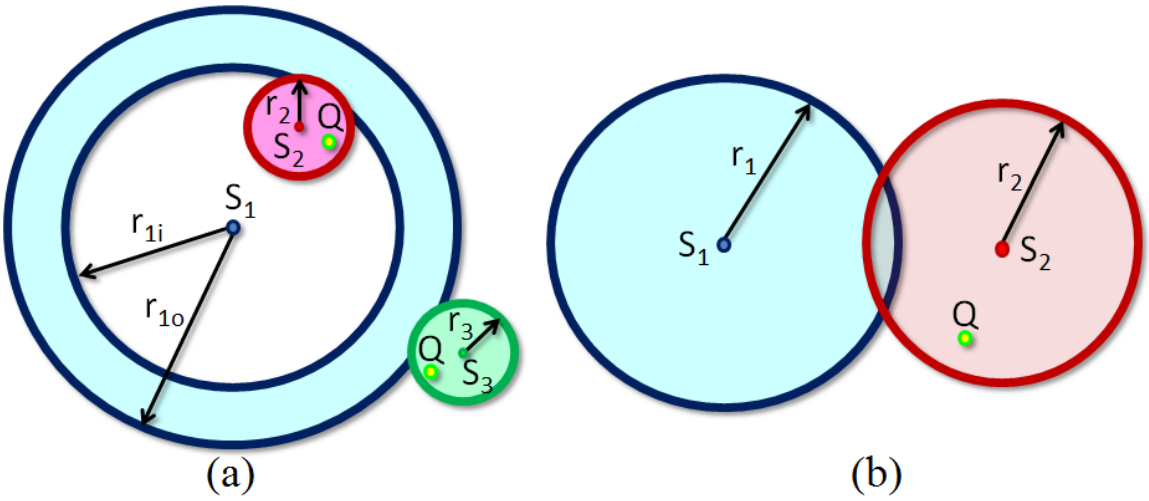
Obrázek 2.46: Ohodnocení pozice podle vzdálenosti vzorku od (a) rotační osy, (b) roviny kolmé k ose rotace. (a) Model řezů. (b) Slupkový model.

M, kdežto pro hraniční roviny je odlišná šířka řezu směrem ven a dovnitř obálky součásti. Hodnoty ohodnocení řezů v modelu řezů jsou relativně vůči sobě porovnány na obrázku nahoře. Všechny parametry definující model řezů jsou zadány relativně ke zvětšené výšce obálky součásti. Na obrázku 2.47 uprostřed je znázorněn příklad ohodnocení viditelných vzorků součástí podle modelu řezů a jejich vytvořené statické ilustrace se šipkami.

Účelem ohodnocení vzorku podle slábnutí je zamezit kumulaci šipek. Před vykreslením každé součásti soustavy jsou známy pozice všech doposud vykreslených šipek pro součásti soustavy. Pro každou doposud vykreslenou šipku se vypočte vzdálenost pozice této šipky (viz obrázek 2.43(a,b) bod M) ke všem viditelným vzorkům součásti, která má být vykreslena. Vypočtené vzdálenosti se normalizují největší vzdáleností přes všechny vzdálenosti od dané již vykreslené šipky k viditelným vzorkům analyzované součásti. Poté se výpočet opakuje pro všechny doposud vykreslené šipky a normalizované vzdálenosti od doposud vykreslených šipek se kumulují pro každý viditelný vzorek. Nakonec dojde k podílu ohodnocení vzorku z hlediska slábnutí počtem vykreslených šipek. Čím menší je vypočtená vzdálenost dvojice vzorků, tím menší je ohodnocení tohoto vzorku z hlediska slábnutí. Nejlepší ohodnocení z hlediska slábnutí je pro nejvzdálenější dvojici vzorků. Výpočet vzdálenosti pro účel zabránění kumulace šipek se provádí pro 2D, protože chceme zabránit tomu, aby docházelo ke kumulaci šipek ve snímku. Příklad ohodnocení vzorků soustavy součástí, která je složena ze tří součástí, z hlediska slábnutí je znázorněn na obrázku 2.47 dole. Součást, jejíž všechny viditelné vzorky jsou obarveny červenou barvou, přestavuje první součást soustavy, pro kterou byla vygenerována šipka znázorňující pohyb součásti. Pro tuto součást neexistovala doposud žádná vykreslená šipka, a proto ohodnocení všech viditelných vzorků této součásti z hlediska slábnutí je nulové (stejné). Ohodnocení vzorků Pro zelenou součást je ovlivněno umístěním šipky, která znázorňuje rotaci červené součásti. V neposlední řadě ohodnocení viditelných



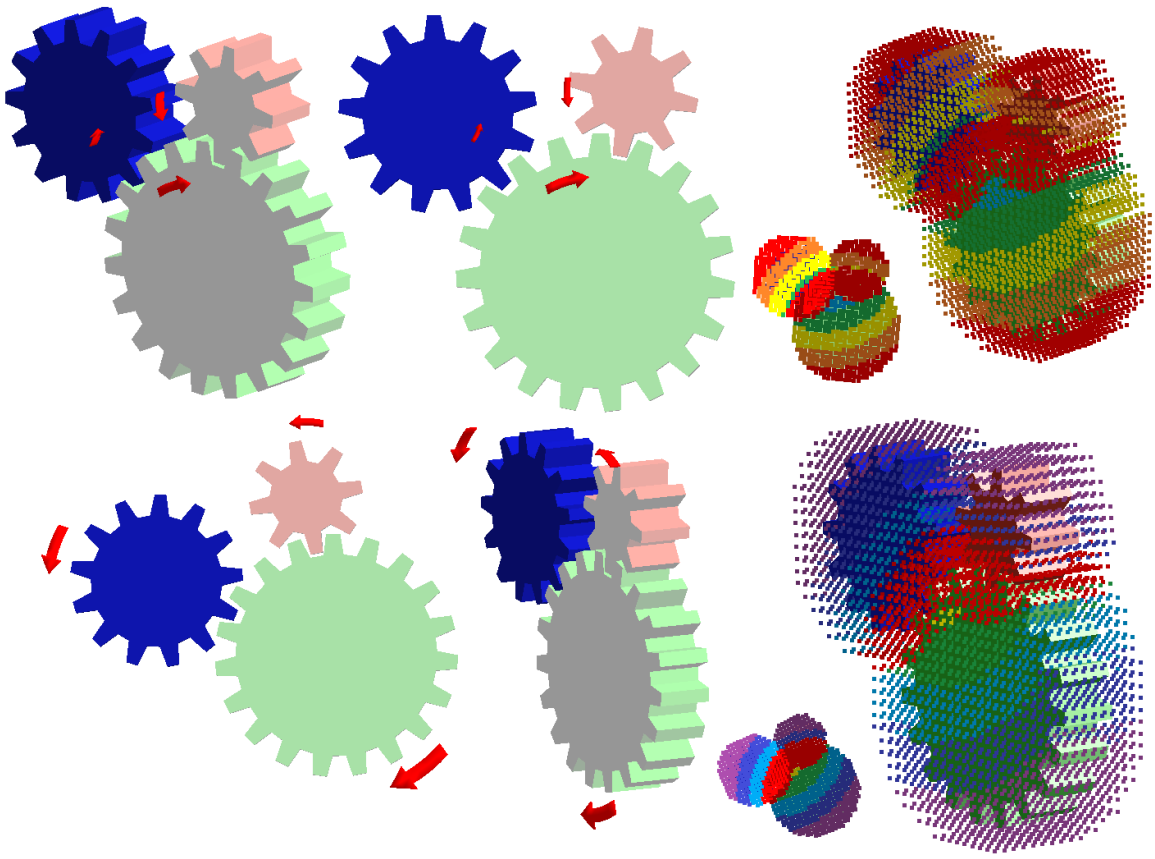
Obrázek 2.47: Ohodnocení vzorků podle (nahoře) slupkového modelu, (uprostřed) modelu řezů a (dole) slábnutí. Porovnání relativních hodnot ohodnocení je následující: fialová > tmavě modrá > světle modrá > zelená > žlutá > oranžová > červená.



Obrázek 2.48: Princip stanovení přitažlivých/odpudivých sil pro spojení součástí se (a) součástí, která má vnitřní a vnější zuby, a (b) součástí, které tuto součást neobsahují.

vzorků modré součásti jsou ovlivněny umístěním šipky, která znázorňuje pohyb červené a zelené součásti.

Ohodnocení viditelných vzorků součástí z hlediska silového modelu je závislé na typu zvolené síly (přitažlivá nebo odpudivá síla), která má být použita mezi takovými součástmi soustavy, které spolu mají kontakt. Pro tento model je tedy důležité znát graf interakce analyzované soustavy. Odpudivá síla představuje inverzní sílu k přitažlivé síle, pro kterou platí analogické principy jako pro přitažlivou sílu, a proto v následujícím textu bude vysvětlen princip silového modelu pouze na přitažlivé síle. Velikost přitažlivé síly je závislá na vzdálenosti viditelného vzorku součásti P_i, pro kterou jsou generovány aktuálně šipky, k ose rotace součásti P_j, která má kontakt se součástí P_i, a na stanovení, zda viditelný vzorek leží uvnitř zvětšené obálky součásti P_j. Zjištění, zda vzorek součásti P_i leží uvnitř zvětšené obálky součásti P_j, se určí podle vzdálenosti vzorku od osy rotace součásti P_j a podle vzdálenosti vzorku od roviny, ve které leží těžiště součásti P_j, a jejíž normálový vektor je totožný s osou rotace součásti P_j. Pokud vzorek leží uvnitř součásti P_j, pak hodnota přitažlivé síly je rovna nule, tzn. jedná se o nejhorší ohodnocení. V této práci je možno nastavit tzv. tvrdý mód. Tvrdý mód je takový mód, ve kterém je výsledné ohodnocení vzorku rovno nulovému ohodnocení, pokud leží vzorek uvnitř zvětšené obálky součásti P_j. Možnost použít tvrdý mód byla přidaná z důvodu, aby nedocházelo k takovému umístění šipek při působení přitažlivých sil, při kterém by byla zvolená taková pozice šipky, která by ležela uvnitř zvětšené obálky součásti P_j. Stanovení velikosti přitažlivé síly se liší pro součást s vnějšími a vnitřními zuby oproti ostatním součástem soustavy (viz obrázek 2.48). Pro spojení součástí, které neobsahuje součást s vnějšími a vnitřními zuby, platí, že čím je menší vzdálenost vzorku k ose rotace součásti P_j, tím je větší přitažlivá síla (viz obrázek 2.48(b)). Pokud spojení obsahuje součást P_i, která má vnitřní a vnější zuby, a pokud je součást P_j vně součásti P_i, pak je princip pro přitažlivou sílu stejný. Pokud však je součást P_i uvnitř součásti P_j, která má vnitřní a vnější



Obrázek 2.49: Ohodnocení vzorků podle silového modelu (nahoře) přitažlivé síly a (dole) odpudivé síly. Porovnání relativních hodnot ohodnocení je následující: fialová > tmavě modrá > světle modrá > zelená > žlutá > oranžová > červená.

zuby (viz obrázek 2.48(a)), pak přitažlivá síla roste se zvětšující se vzdáleností vzorku od osy rotace součásti P_j. Příklady vytvořených statických ilustrací se šipkami s aplikováním přitažlivých a odpudivých sil jsou znázorněny na obrázku 2.49.

Pro umístění šipky znázorňující pohyb součásti je zvolen vzorek s největším výsledným ohodnocením. Výsledné hodnoty ohodnocení vzorků součásti závisí na velikosti faktorů jednotlivých kritérií výsledného ohodnocení. Jednotlivé klíčové kroky pro stanovení umístění šipky pro pohybující se součást jsou uvedeny v algoritmu 3. Princip, jak zarovnat šipku do scény do konkrétního vzorku, je popsán v algoritmu 4.

Souřadnicové systémy šipek, které jsou použity v této práci, jsou zobrazeny na obrázku 2.43. Osou rotace všech šipek je souřadnicová osa z. Vzdálenost středu všech typů šipek, který je znázorněn na obrázku 2.43 jako bod M, je rovna jedné jednotce ve směru souřadnicové osy y. Souřadnicové osy lokálního souřadnicového systému šipky jsou znázorněny na obrázku 2.43 jako osa x, y a z. Počátek lokálního souřadnicového systému šipky je označen jako bod O a má souřadnice [0, 0, 0]. Šipky, které se umísťují na podstavu objektu (z angl. cap arrow), jsou znázorněny na obrázku 2.43(c) a 2.43(e). Šipky, které se umísťují na plášť objektu (z angl.

side arrow), jsou znázorněny na obrázku 2.43(d) a 2.43(f). Rozdíl mezi dvojicí šipek, které se umísťují na stejnou část objektu (podstavu nebo plášť), je, že první zmíněná šipka má střed (bod M) uprostřed délky šipky a existuje pro ní bod T (z angl. tail), který je označen jako ocásek šipky, kdežto druhá zmíněná šipka má střed na hraně šipky a neexistuje pro ní bod T. Šipku, která má střed uprostřed délky šipky, je vhodné umístit pro takové objekty, které mají větší plochu podstavy nežli je celková délka šipky, aby nedocházelo k tomu, že bude bod T mimo součást. Naopak šipku, která má střed na své hraně, je vhodné umístit pro objekty, které mají menší plochu podstavy, pro spojení kuželové ozubené soukolí a spirálové ozubené soukolí, která znázorňuje rotaci dynamické součásti okolo rotační osy nepohyblivé součásti. Pro tyto dva typy spojení je vhodnější tato kratší šipka, protože nebude zbytečně zakryta její část od středu ke konci šipky objektem. Kratší typ šipky je také vhodné použít pro situaci, kdy jsou součásti umístěny blízko u sebe a je potřeba znázornit pohyb dynamické součásti okolo rotační osy nepohyblivé součásti. V následujícím textu bude používán výraz hlavička (bod H), ocásek (bod T) a střed (bod M) šipky, které jsou znázorněny na obrázcích 2.43 dole. V této práci je podporována také možnost umístit šipky, které znázorňují rotaci dynamické součásti okolo rotační osy nepohyblivé součásti, do průsečíků osy rotace součásti s rovinami, ve kterých leží podstavy zvětšené obálky součásti. Tato možnost byla přidána z důvodu, že tato šipka bude překrývat méně ostatních objektů ve scéně. Pokud je zvoleno vykreslit šipku do středu součásti, aby šipka znázorňující rotaci dynamické součásti okolo rotační osy fixní součásti zakrývala pokud možno co nejméně ostatních objektů ve scéně, pak pro součást, která má vnitřní a vnější zuby nemůže být tato šipka vykreslena do jejího středu, protože tato součást nemá plochu ve svém středu, ale vypočte se směrový vektor ze středu součásti do nalezeného průsečíku dvojice součástí, které spolu mají kontakt, a z poloviční vzdálenosti rozdílu vnitřního a vnějšího poloměru součásti se posune tento průsečík ve směru stanoveného směrového vektoru o definovanou hodnotu rozdílu.

Algoritmus 3 Vytvoř šipku pro pohyblivou součást

1: Vytvoř zvětšenou válcovou/ kuželovou obálku BB součásti.
2: Vytvoř vzorky uvnitř BB.
3: Ohodnoť vytvořené vzorky.
4: Zjisti, jaké vzorky jsou viditelné ve scéně.
5: **StanovPoziciŠipky(vzorek)**.
6: Vykresli natočenou šipku pro každý viditelný vzorek.
7: Zjisti počet viditelných fragmentů šipky.
8: Najdi takový vzorek a pro něho druh šipky, které mají nejlepší celkové ohodnocení.
9: Vykresli výslednou šipku pro pohyblivou součást.

Pouze dynamická součást může rotovat jak okolo své osy rotace tak okolo osy rotace nepohyblivé součásti. Stanovení umístění šipky pro dynamickou součást, která je nucena rotovat okolo rotační osy nepohyblivé součásti, je provedeno dle vzájemné polohy rotačních os a tvarů dvojice objektů, které spolu mají kontakt. Součást, která je označena na obrázku 2.50 jako P_1, je nepohyblivou součástí a součást, která je označena jako P_2, je dynamickou součástí. Na obrázku 2.50 jsou znázorněny konfigurace párů součástí, na kterých jsou zobrazeny principy pro nalezení umístění šipky znázorňující pohyb dynamické součásti okolo nepohyblivé součásti a pro stanovení, jaká nalezená pozice je vhodná pro jaký směr šipky. Body I_1, I_2 představují nalezené pozice pro umístění šipky, která znázorňuje rotaci dynamické sou-

Algoritmus 4 StanovPoziciŠipky(vzorek)

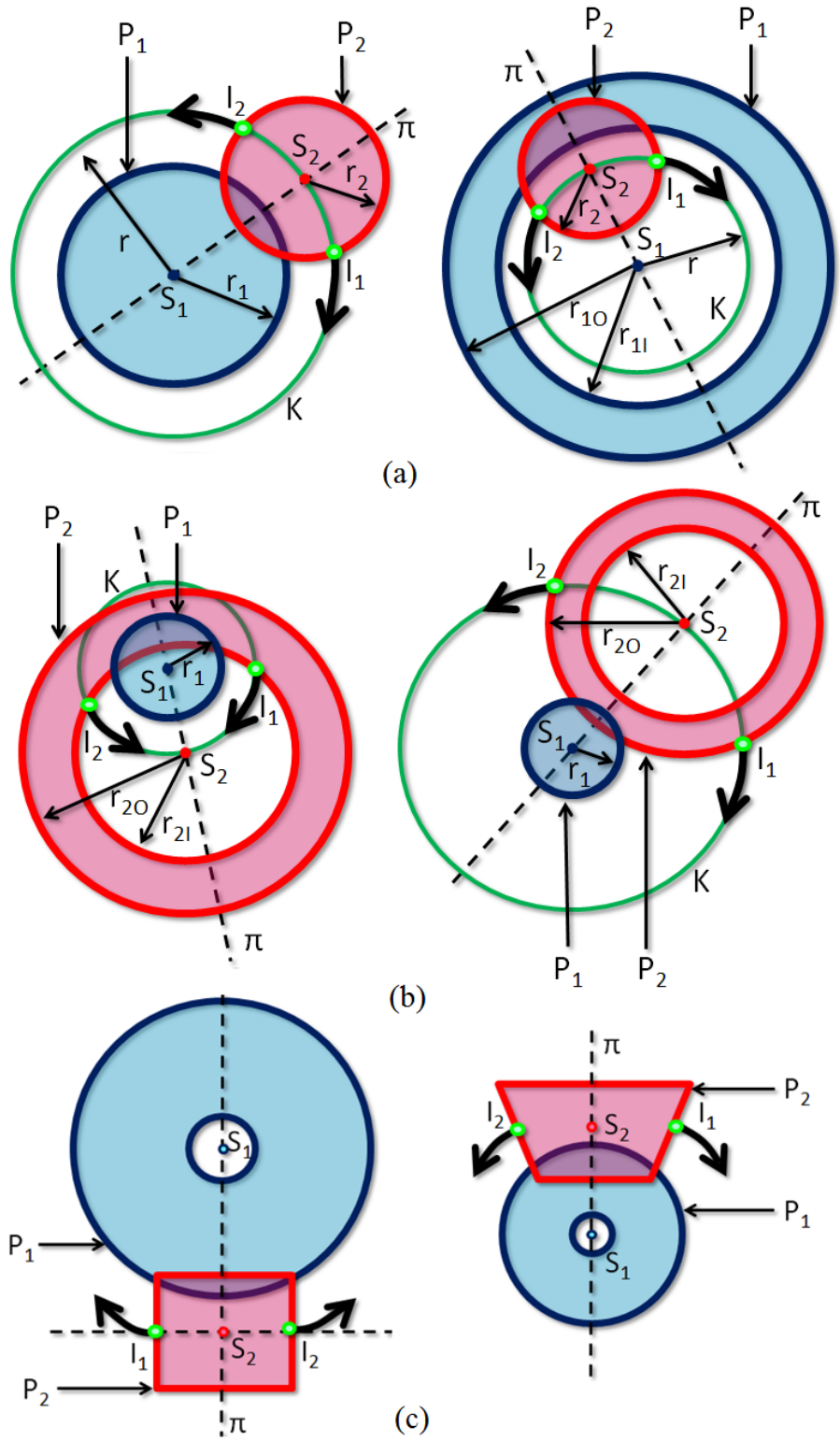
Vstup: vzorek uvnitř zvětšené obálky

1: Vypočti transformační matici pro zarovnání LCS součásti do GCS a ulož si její inverzi.
2: Vypočti úhel rotace šipky okolo osy$-y$ GCS podle směru šipky.
3: Vypočti úhel rotace pro zarovnání rotační osy šipky s rotační osou součásti.
4: Vypočti poloměr šipky, který je roven vzdálenosti od počátku GCS ke vzorku.
5: Vypočti úhel rotace, který otočí šipku okolo rotační osy součásti tak, aby střed šipky (bod M) ležel v souřadnicích vzorku.
6: Vypočti posun pozice šipky podél rotační osy součásti na souřadnici vzorku.

části okolo rotační osy nepohyblivé součásti. Rovina π je určena dvěma směrovými vektory a bodem, který leží v této rovině. Směrovým vektorem roviny π je osa rotace nepohyblivé součásti a směrový vektor spojující počátky lokálních souřadnicových systémů součástí. Počátky lokálních souřadnicových systémů obou součástí leží v rovině π. Princip nalezení působiště šipky, která znázorňuje rotaci dynamické součásti okolo nepohyblivé součásti, pro dvojici součástí, které spolu mají kontakt, jejichž rotační osy jsou rovnoběžné, je založen na nalezení průsečíků kružnice dynamické součásti P_2 (červená kružnice) a kružnice K (zelená kružnice), která má střed totožný se středem těžiště nepohyblivé součásti a jejíž poloměr je roven vzdálenosti transformované pozice těžiště nepohyblivé a dynamické součásti do roviny ρ. Vytvoří se rovina ρ, jejíž směrové vektory jsou bázovými vektory buď součásti P_1 nebo P_2 a které nejsou osou rotace dané součásti. V rovině ρ leží počátek lokálního souřadnicového systému součásti, jejíž bázové vektory jsou směrovými vektory této roviny. Kružnice dynamické součásti P_2 je definována tak, že střed kružnice je projekcí počátku lokálního souřadnicového systému součásti (těžiště) P_2 do roviny ρ, a jejíž poloměr je roven poloměru součásti P_2. Existence dvou průniků dvojice kružnic se předpokládá z důvodu, že dvojice součástí, které spolu mají kontakt, je do sebe dostatečně zaklesnuta, aby docházelo k tomu, že rotace jedné součásti vyvolá rotaci součásti, která s ní má kontakt.

V této práci je požadováno, aby hlavička šipky znázorňující rotaci dynamické součásti okolo rotační osy nepohyblivé součásti směřovala vždy směrem ze součásti. Je nutné stanovit, jakým směrem má být orientována šipka, jejíž střed (bod M) bude umístěn do nalezeného průsečíku I_1 a I_2, aby byla splněna podmínka, že hlavička šipky bude směřovat směrem ven ze součásti. Jak je z patrné z obrázku 2.50, tak pro všechny spojení a pro libovolnou záměnu nepohyblivé a dynamické součásti až na jednu výjimku musí pro správně orientovanou šipku platit, že střed šipky leží blíže k rovině π než hlavička šipky. Výjimkou je případ, který je znázorněn na obrázku 2.50(b) vlevo a pro který musí pro správně orientovanou šipku platit, že hlavička šipky je blíže k rovině než střed šipky. Pro správnou orientaci šipky zároveň musí platit pro všechny typy spojení a také pro uvedenou výjimku, že hlavička a střed šipky musí ležet na jedné straně od roviny π.

Pokud bude osa rotace dynamické a nepohyblivé součásti totožná, pak nedojde k vygenerování šipky, která by znázorňovala rotaci této dynamické součásti okolo rotační osy nepohyblivé součásti, protože se dynamická součást stane statickou součástí. Proto nemůže nastat situace, že by vytvořená kružnice K (viz na obrázku 2.50 zelená kružnice), která má střed v počátku lokálního souřadnicového systému nepohyblivé součásti P_1 a která má po-

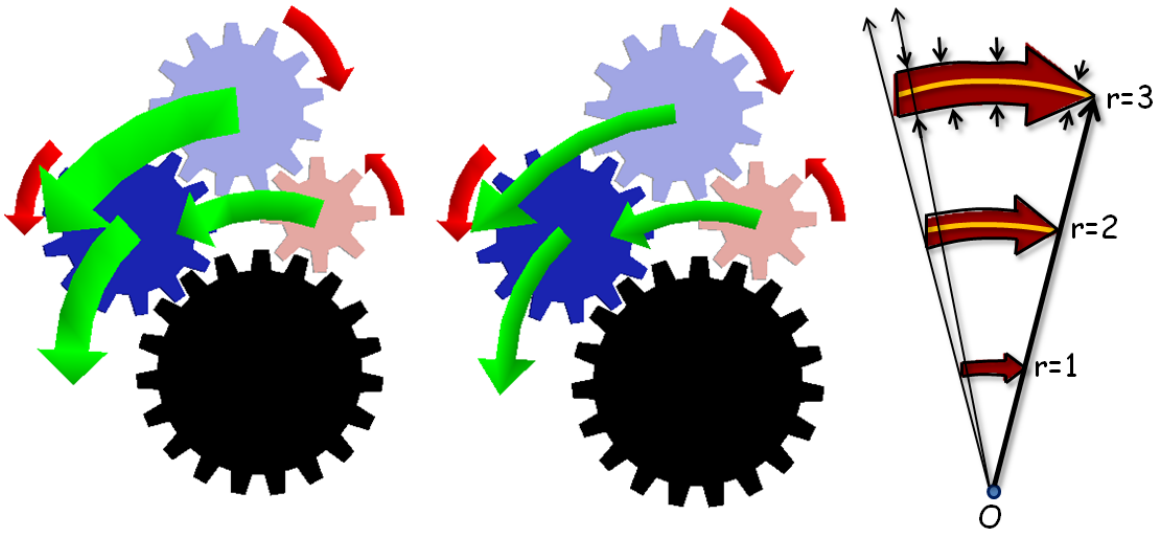


Obrázek 2.50: Princip stanovení pozice šipky, která znázorňuje rotaci dynamického objektu okolo rotační osy nepohyblivé součásti, pro rovnoběžné osy a kolmé osy pro spojení (c) vlevo spirálové ozubené soukolí a (c) vpravo kuželové ozubené soukolí. P_1 – nepohyblivá součást; P_2 – dynamická součást

loměr roven vzdálenosti osy rotace nepohyblivé a dynamické součásti, měla nulový poloměr. Poloměr kružnice dynamické součásti P_2, jejíž průnik s kružnicí K je požadováno nalézt, odpovídá poloměru součásti P_2. Pokud však je dynamickou součástí válec, který má vnitřní a vnější poloměr (viz obrázek 2.50(b) součást P_2), pak je potřeba rozhodnout, zdali bude použit pro nalezení pozice šipky její vnější nebo vnitřní poloměr. Pouze pro případ, který je znázorněn na obrázku 2.50(b) vlevo, bude použit pro stanovení pozice šipky znázorňující rotaci dynamické součásti P_2 okolo rotační osy nepohyblivé součásti P_1, vnitřní poloměr součásti P_2 na místo jeho vnějšího poloměru. Na obrázku 2.50(b) vlevo je znázorněna soustava dvojice součástí, které spolu mají kontakt, která tvoří výjimku, pro kterou je stanoveno jiné kritérium pro správnou orientaci šipky, která je umístěna do stanovené pozice, a pro kterou platí, že nepohyblivá součást P_1 má menší poloměr a leží uvnitř dynamické součásti P_2, která má odlišný vnitřní (r_{2I}) a vnější (r_{2O}) poloměr. Pokud by byl pro nalezení pozice šipky použit pro stejnou konfiguraci součástí, která je znázorněna na obrázku 2.50(b) vlevo, vnější poloměr součásti P_2, pak by došlo buď k tomu, že pokud by byla orientace šipek totožná s vykreslenými šipkami v tomto obrázku, pak by hlavička šipky směrovala dovnitř součásti, a nebo by došlo k tomu, že pokud by se orientace šipek znázorněných na tomto obrázku obrátila, pak by sice platilo, že hlavička šipky by směřovala ven ze součásti a že hlavička šipky by byla blíže k rovině π než střed šipky, ale takto umístěná šipka by byla velice špatně viditelná a esteticky by toto umístění šipky bylo horší nežli požadované vykreslené umístění šipek, které je dle mého názoru estetičtější.

Pokud je osa rotace nepohyblivé součásti kolmá na osu rotace dynamické součásti, pak bude pozice šipky, která bude znázorňovat rotaci dynamické součásti okolo osy rotace nepohyblivé součásti, nalezena podle principu, který je znázorněn na obrázku 2.50(c) a který bude nyní popsán. Princip nalezení umístění šipky pro součásti, které mají kolmé osy rotace, závisí na tvaru součásti. Umístění šipky, která znázorňuje rotaci dynamické součásti okolo rotační osy nepohyblivé součásti, pro součást, která má tvar válce, je takový bod, který je průsečíkem osy rotace součásti s rovinou, ve které leží podstava součásti (viz obrázek 2.50(c) vlevo). Princip nalezení pozice šipky, která znázorňuje rotaci dynamické součásti P_2 okolo rotační osy nepohyblivé součásti P_1, pro součásti, které mají tvar komolého kužele, je znázorněn na obrázku 2.50(c) vpravo. Pozice šipky pro součást, která má tvar komolého kužele, odpovídá bodu, který leží na plášti kužele a jehož souřadnice jsou $B = \vec{v} \cdot r + \vec{O}$, kde r je poloměr součásti v daném řezu, $\vec{O}$ je počátek lokálního souřadnicového systému (těžiště) dynamické součásti a $\vec{v}$ je normalizovaný vektor, který je kolmý k ose rotace nepohyblivé a dynamické součásti. Protože poloměry podstav komolé kužele nejsou stejné, tak je potřeba pro zjištění libovolné hodnoty poloměru v požadované výšce komolého kužele lineárně interpolovat hodnoty poloměrů podstav. Bod, který odpovídá umístění šipky, leží v rovině, ve které leží těžiště součásti, tzn. tento bod leží ve stejné vzdálenosti od obou podstav komolého kužele.

Byl popsán princip nalezení dvou pozic pro umístění šipky, která znázorňuje rotaci dynamické součásti okolo rotační osy nepohyblivé součásti. Jedna z dvojice vygenerovaných pozic je vhodná pro umístění šipky, která znázorňuje směr rotace součásti ve směru hodinových ručiček, a druhá vygenerovaná pozice je vhodná pro umístění šipky znázorňující rotaci součást proti směru hodinových ručiček. Pro dvojici součástí, jejichž osy rotace jsou rovnoběžné, se vytvoří tří potencionálních pozice pro každý směr šipky pro pozici šipky, která byla stanovena dle popsaného postupu v nulové vzdálenosti od roviny π, jejíž normálový vektor



Obrázek 2.51: Soustava součástí s (vlevo) neregulovanou výškou šipek a (uprostřed) zelené šipky mají stejnou výšku. Vpravo je znázorněn princip extrudování a zvětšení šipky.

je osa rotace součásti a v níž leží těžiště součásti. Další dvě pozice pro stejný směr šipky leží v rovinách, ve kterých leží podstavy zvětšené obálky součásti. Kvůli estetickému hledisku došlo k vygenerování těchto tří vzorků pouze pro součásti, které mají rovnoběžné osy rotace.

Pro šipky znázorňující rotaci dynamické součásti okolo rotační osy nepohyblivé součásti byla použita technika dotazů na zastínění pouze pro stanovení počtu viditelných fragmentů šipky. Nebylo žádoucí eliminovat potencionálních působiště těchto šipek, které nejsou viditelné, kvůli malému množství těchto působišť. Pro každé vygenerované působiště se vykreslí oba dva typy šipek, které jsou podporovány v této práci. Poté dojde ke stanovení takového působiště a typu šipky, pro kterou existuje největší počet viditelných fragmentů šipky. Pokud je počet viditelných fragmentů šipky, která má být vykreslena, větší nežli stanovená mez pro minimální počet viditelných fragmentů šipky, pak je šipka vykreslena. V opačném případě nebude vykreslena pro danou součást žádná šipka, která by znázorňovala rotaci dynamické součásti okolo rotační osy nepohyblivé součásti.

V případě, když by byly objekty blízko u sebe, měly by malou viditelnou plochu a byly vzdáleny daleko od nepohyblivé součásti, okolo jejíž osy rotace by byly donuceny rotovat, tak by došlo k tomu, že výška této šipky by byla příliš velká a docházelo by k tomu, že by šipka překrývala ostatní šipky a také objekty v její blízkosti. K tomuto chování by docházelo z důvodu zvětšení šipky na pozici vzorku. Aby byla statická ilustrace se šipkami přehlednější, tak byla do této práce přidána možnost nastavit pevně danou výšku všech šipek, které znázorňují rotaci dynamické součásti okolo nepohyblivé součásti. Popisovaná situace je znázorněna na obrázku 2.51. Pro šipky, které znázorňují rotaci objektu okolo vlastní rotační osy, nebyla použita regulace výšky šipky, protože velikost těchto šipek lze měnit dle vhodnosti ohodnocení vzorku podle vzdálenosti vzorku od osy rotace součásti. Vypočte se faktor extrudování vrcholu šipky ze znalosti poloměru, výšky a tloušťky namodelované šipky. Je známa

souřadnice středu šipky a bodu, který leží na kontuře šipky. Faktor extrudování je roven $e = (r+h)\cdot\ d - r\cdot\ d - h$, kde r je poloměr šipky (1 jednotka), h je tloušťka šipky (hodnota 0.02 jednotky pro typ šipky, který se umísťuje na plášť součásti) nebo výška šipky (hodnota 0.1 jednotky pro typ šipky, která se umísťuje na podstavu součásti) a d je vzdálenost vzorku od osy rotace součásti. Transformace souřadnice vrcholu šipky probíhá stejným způsobem jako zarovnání pozice středu šipky do souřadnic vzorku. Stanoví se normalizovaný směrový vektor, který směřuje od bodu O k transformované pozici vrcholu šipky. Bod O představuje průsečík roviny, jejíž normálový vektor je osa rotace součásti, a v níž leží těžiště součásti, a osy rotace součásti. Vzdálenost r_s představuje vzdálenost vzorku, do kterého má být umístěn střed šipky, od osy rotace součásti. Pokud je vzdálenost analyzovaného vrcholu šipky od osy rotace součásti větší nebo menší nežli r_s a zároveň je r_s větší nežli poloměr šipky, pak dojde k posunu tohoto vrcholu směrem dovnitř šipky. Pokud je hodnota r_s menší nežli poloměr šipky (1 jednotka), pak dojde k posunu vrcholů šipky opačným směrem nežli bylo popsáno. Popsaným postupem lze docílit stejně vysoké šipky.

2.6 Shrnutí

Pro vytvoření dynamické vizualizace a statické ilustrace se šipkami soustavy mechanických součástí bylo potřeba zjistit vlastnosti součástí soustavy a stanovit, jaké součásti soustavy mají spolu kontakt. Vlastnosti součásti určily, jak se bude chovat součást v soustavě. Vlastnosti součásti, jimiž byla osa rotace, výška, poloměr podstav, počet zubů a tvar, byly nalezeny aplikováním principu metody analýzy hlavních komponent, nalezením extrémních vrcholů konvexní obálky a vytvořením shluků. Páry součástí soustavy, které měly spolu kontakt, byly nalezeny aplikováním metody detekce kolizí s využitím voxelizace objektu na extrudované objekty. Součásti soustavy, které měly spolu kontakt, bylo potřeba znát, aby bylo možné stanovit šíření kinetické energie soustavou směrem od řídící součásti. Byl sestrojen graf interakce soustavy, který byl vytvořen na základě znalosti vlastností součástí a párů součástí soustavy, které měly spolu kontakt. Na základě průchodu grafu interakce a znalosti vlastností součástí byly stanoveny parametry animace součástí soustavy, pokud byl proveditelný pohyb součástí soustavy. Parametry rotace dvojice součástí, které měly spolu kontakt, se vypočetly ze znalosti rychlosti rotace řídící součásti a počtu zubů řízené a řídící součásti. Pokud existovala mezi analyzovanou dvojicí součástí hrana v grafu interakce a řídící součástí byla součást P_1 a řízená součást byla součást P_2, pak byl výpočet rychlosti rotace součásti P_2 roven $\omega_{P_2} = \omega_{P_1} \cdot (N_{P_1}/N_{P_2})$, kde N_{P_i} byl počet zubů součásti P_i a ω_{P_i} představoval úhlovou rychlost rotace součásti P_i. V této práci byly podporovány pomocné součásti, které sloužily jako kontext pro pochopení funkce soustavy. Pro soustavu součástí, pro kterou byl možný pohyb, došlo k vytvoření statické ilustrace se šipkami. Nalezení umístění šipky součásti bylo provedeno použitím slupkového modelu, modelu řezů, slábnutí a silového modelu. Slupkový model definoval vzdálenost vzorku od osy rotace součásti. Model řezů definoval vzdálenost vzorku podél rotační osy součásti. Ohodnocení z hlediska slábnutí zabraňovalo shlukování šipek a bylo založeno na vypočtení vzdáleností vzorků dvojice součástí ve 2D (pro snímek). Silový model realizoval buď přitažlivý nebo odpudivý efekt, který byl aplikován na takové dvojice součástí, které spolu měly kontakt.

Analyzovaná součást musela mít buď tvar válce nebo komolého kužele, nesměla být deformována a její obálka musela mít tvar válce nebo komolého kužele. Tvar součásti musel

být takový, že vrcholy, které ležely na podstavě součásti, ležely v takové rovině, která byla rovnoběžná s rovinou, jejíž normálový vektor byl shodný se směrovým vektorem osy rotace součásti, a v níž ležel počátek lokálního souřadnicového systému součásti. Pro určení správného počtu zubů součásti, musela být zadána korektní hodnota počtu extrémních vrcholů, které tvořily jeden zub součásti. Aby byl správně určen počet vnitřních zubů součásti, která měla vnitřní a vnější zuby, bylo nutné, aby vrcholy tvořící povrch součásti, které měly nejmenší vzdálenost ze všech jejích vrcholů od její osy rotace, ležely pouze na jejích ostrých hranách. Omezením na detekci kolizí bylo, že pokud pro analyzovanou dvojici součástí neexistovala kolize jejich vrcholů s jejich objemovými daty a protínaly se pouze hrany (stěny) objektů, pak nebylo možné pro takovou dvojici součástí nalézt jejich kontakt. Omezením na vytvoření grafu interakce bylo, že se muselo jednat o jedno z podporovaných spojení součástí, tzn. vzájemná poloha rotačních os součástí musela být v definovaném rozsahu, jinak nedošlo ke správnímu určení spojení součástí. Množina součástí, s nimiž měla součást kontakt, se nesměla během celé animace měnit. Pokud byl poměr počtu zubů řízených a řídících součástí na cestě mezi řídící součástí a nepohyblivou součástí roven jedné a směr rotace obou součástí byl stejný, pak nešlo provést animaci analyzované soustavy z důvodu omezení použité metody. Řídící součástí soustavy mohla být pouze jedna její součást, která nemohla být pomocnou součástí.

Kapitola 3

Realizace

Navržená aplikace byla implementována v jazyce C++ s pomocí několika knihoven. Pro analýzu hlavních komponent (PCA) mechanických součástí byla použita knihovna JAMA. Pro výpočet konvexních obálek mechanických součástí byla použita knihovna CGAL. Pro vykreslení mechanických součástí byly použity knihovny OpenGL, GLEW, GLM a GLUT. Pro vykreslení uživatelského rozhraní byla použita knihovna AntTweakBar. Více informací o použitých knihovnách je možné najít v příloze A.

V této kapitole budou popsány vstupní data, parametry příkazové řádky a požadovaná struktura souboru definující typy součástí a souboru definujícího seznam viditelných pomocných součástí aplikace realizované v této práci. V následujícím textu bude popsáno, v jakých třídách jsou realizovány jednotlivé části zadání a jaké jejich funkce obsahují hlavní princip řešení jednotlivých částí zadání, který byl popsán v analýze a návrhu.

Podrobná dokumentace jednotlivých parametrů metod, popis jejich funkce a popis tříd je uveden v podadresáři *doc* adresáře *realizace* na přiloženém CD, ve které je umístěna dokumentace zdrojových kódů této práce.

Vstupní data, parametry příkazové řádky a požadovaná struktura souborů V souboru s definicí scény je definována topologie každého modelu, který představuje součást soustavy (souřadnice vrcholů, jaké vrcholy tvoří jaké stěny, normálový vektor a index vrcholu/index texturovací souřadnice/index normály). Model je popsán pomocí povrchové reprezentace, jehož povrch je tvořen trojúhelníkovou sítí. Definice materiálů součástí je v souboru s příponou *mtl*. Jméno souboru s definicí materiálů a jméno souboru definicí scény musí být stejná a umístěná ve stejné úrovni adresáře. Definice scény byla zpracována použitím knihovny Assimp[3].

Definice součástí soustavy, typů součástí a informace o viditelných pomocných součástech je zpracována ve třídě *Scene* aplikace realizované v této práci. Mechanická součást soustavy je realizována v této práci třídou *MechPart*, která obsahuje informace o součásti (např. parametry animace, vlastnosti). Konfigurační soubor aplikace představuje třída *Config*, která obsahuje nastavitelné vlastnosti programu, jimiž jsou iniciální hodnoty pro výpočet, meze pro výpočet, iniciální hodnoty pro GUI aplikace, umístění shaderů, umístění definice šipek.

Realizovaná aplikace očekává tři vstupní parametry příkazové řádky, kde první obsahuje relativní umístění definice scény (soubor ve formátu *obj*), druhý obsahuje relativní umístění souboru definujícího typy součástí (soubor ve formátu *txt*) a třetí představuje relativní

umístění souboru definujícího seznam viditelných pomocných součástí soustavy (soubor ve formátu *txt*). První uvedený parametr s definicí scény je povinným parametrem a ostatní dva parametry jsou nepovinné.

Relativní umístění definice topologie šipek použitých pro vytvoření statické ilustrace je zadáno ve třídě *Config*. Proměnná *arrowFrontFileName* představuje cestu k 3D modelu šipky, která se umísťuje na podstavu objektu, a proměnná *arrowSideFileName* přestavuje cestu k 3D modelu šipky, která se umísťuje na plášť součásti. Umístění shaderů pro voxelizaci objektu je v proměnné *voxelVsFile* je umístění vertex shaderu, a v proměnné *voxelFsFile* je umístění fragment shaderu pro voxelizaci mechanických součástí při detekci kolizí. Umístění shaderů pro detekci kolizí vrcholů testovaného objektu s objemovými daty voxelizovaného objektu, kde proměnná *collisionVsFile* obsahuje relativní umístění vertex shaderu a proměnná *collisionFsFile* obsahuje relativní umístění fragement shaderu. Fragment shader pro nastavení hloubky fragmentu pro vykreslení šipky je v proměnné *arrowFsFile*. Hloubku vykreslené šipky bylo potřeba nastavit z důvodu, aby nedocházelo k překreslování již vykreslených šipek soustavy.

Požadovaná struktura souboru s definicí typů součástí soustavy je následující:

1. pokud není součást pomocnou součástí: „Pi T“,
2. pokud je součást pomocnou součástí: „Pi T Pj“,

kde $i \in \mathcal{N}$, představuje index součásti soustavy, jehož typ bude definován, a který odpovídá pořadí součásti definované v souboru s definicí scény. Indexování součástí začíná indexem nula. Symbol T představuje typ součásti, který může nabývat jedné z následujících hodnot D/S/F/A, kde D je dynamická součást, S je statická součást, F je fixní (nepohyblivá) součást a A je pomocná součást. Symbol j představuje index součásti soustavy, s níž bude mít pomocná součást stejný pohyb, a nabývá buď hodnot z množiny přirozených čísel N a nebo má hodnotu -1, která značí, že je pomocná součást nepohyblivá. Symbol $\mathcal{N}$ představuje množinu přirozených čísel, která obsahuje i hodnotu 0.

Požadovaná struktura souboru s definicí viditelných pomocných součástí je, že na každém řádku bude uveden symbol Pi, kde i je index součásti, který je shodný s pořadím v definici scény, kde indexování součástí začíná od nuly.

3.1 Nalezení vlastností součásti

Požadovanými parametry součásti, které je potřeba znát, jsou osa rotace, poloměr podstav, počet zubů, výška a tvar součásti. Jedna z os lokálního souřadnicového systému součásti je osou rotace součásti. Aby bylo možné určit osu rotace součásti, bylo potřeba vypočítat lokální souřadnicový systém součásti, jehož jedna osa je osou rotace součásti. Pro správné určení rotační osy součásti, stanovení typu součásti a stanovení počtu zubů součásti bylo potřeba uvažovat pro výpočet pouze takové vrcholy, které ležely na podstavách součásti a které ležely na ostrých hranách součásti. Za tímto účelem byla vytvořena konvexní obálka součásti, na níž ležely požadované vrcholy, a byly v této množině nalezeny její extrémní vrcholy. Poté byly vytvořeny shluky pro nalezení rotační osy součásti.

Určení lokálního souřadnicového systému součásti. Lokální souřadnicový systém součásti byl nalezen pomocí metody analýzy hlavních komponent. Funkce určení lokálního souřadnicového systému součásti je realizována ve třídě *PCA*. Nejprve byl vypočten počátek lokálního souřadnicového systému součásti a stanovena kovarianční matice součásti (metoda *setCovMatrix*). Kovarianční matice slouží jako vstup pro metodu, která vypočítá vlastní vektory a vlastní čísla kovarianční matice, které představují bázové vektory lokálního souřadnicového systému součásti. Pro výpočet vlastních vektorů a vlastních čísel regulární matice je v této práci použita knihovna TNT a JAMA[23] a konkrétně její třída *Eigenvalue*. Ve funkci *setEigenVectors* třídy PCA jsou stanoveny vlastní čísla a vlastní vektory kovarianční matice použitím uvedené knihovny JAMA. Lokální souřadnicový systém (bázové vektory a jeho počátek) součásti je uložen ve třídě *LocalCoordSystem*.

Výpočet konvexní obálky a nalezení jejích extrémních vrcholů. Informace o konvexní obálce a operace pro ní prováděné jsou obsaženy ve třídě *ConvexHull*. Pro stanovení, jaký bázový vektor je osou rotace součásti, je potřeba nalézt extrémní vrcholy konvexní obálky součásti. Pro vlastní výpočet konvexní obálky 3D objektu používá třída *ConvexHull* knihovnu CGAL[10] a její funkci convex_hull_3[11]. Funkce convex_hull_3 poskytovaná knihovnou CGAL je implementována tak, že přidá do výsledného konvexního mnohostěnu i vrcholy objektu, které leží na jeho stěnách, tudíž neobsahuje pouze extrémní vrcholy konvexní obálky. Struktura konvexní obálky součásti byla uložena do datové struktury Polyhedron_3[9] knihovny CGAL, který představuje konvexní mnohostěn. Vstupní a výstupní vrcholy objektu pro funkci convex_hull_3 knihovny CGAL jsou typu Point_3, kterou podporuje knihovna CGAL. Aby bylo možné identifikovat, jaké vrcholy z množiny vstupních vrcholů leží na výsledné konvexní obálce, bylo potřeba přidat do třídy Point_3 atribut index. Tento atribut byl nastaven pro všechny vstupní vrcholy tak, aby jeho hodnota odpovídala indexu do pole vrcholů, které tvoří povrch modelu. V metodě *findExtremePoints* třídy *ConvexHull* se stanoví, jaké vrcholy tvořící povrch konvexní obálky jsou jejími extrémními vrcholy.

Vytvoření shluků. V metodě *setRotationalAxis* třídy *PCA* jsou vypočteny vzdálenosti extrémních vrcholů konvexní obálky ke stanoveným bázovým vektorům lokálního souřadnicového systému, je vypočtena mez shluku a stanovena osa rotace součásti. Mez shluku představuje hodnotu rozdílu dvou sousedních hodnot vzdáleností vrcholů k ose lokálního souřadnicového systému součásti v seřazené posloupnosti, při jejímž překročení bude vytvořen nový shluk. Vypočtené vzdálenosti jsou seřazeny a je nalezena taková mez shluku, že existuje pro tuto mez pouze pro jeden bázový vektor jeden nebo dva shluky a pro ostatní bázové vektory musí existovat více shluků.

Bázový vektor lokálního souřadnicového systému součásti, pro něhož existuje jeden nebo dva shluky je osou rotace součásti. Poloměr podstavy součásti je vypočten jako aritmetický průměr hodnot vzdáleností, které jsou obsaženy pro shluk, která představuje analyzovanou podstavu součásti. Počet zubů součásti byl určen jako podíl mohutnosti množiny extrémních vrcholů konvexní obálky součásti hodnotou proměnné *numberCHVerticesTooth*, která je definována v konfiguračním souboru a která přestavuje počet extrémních vrcholů, které tvoří jeden zub. Obecná rovnice tří rovnoběžných rovin, ve kterých leží buď podstavy součásti a nebo počátek lokálního souřadnicového systému součásti, jejichž normálové vektory

jsou shodné s osou rotace součásti, je určena v metodě *determineEquationPlanes* třídy *PCA*. V této metodě je také stanovena výška součásti jako vzdálenost rovin, ve kterých leží podstavy součásti. Stanovení tvaru součásti je realizováno v metodě *determineShape* třídy *PCA* podle počtu shluků a případně podle procentuálního rozdílu poloměrů podstav.

3.2 Nalezení součástí soustavy, které spolu mají kontakt

Nalezení součástí, které spolu mají kontakt je provedeno pomocí voxelizace. Operace pro provádění voxelizace jsou obsaženy ve třídě *Voxelization* a detekce kolizí je ve třídě *Collision*. Shadery, které naleznou objemovou reprezentaci objektu ze znalosti jeho povrchové reprezentace, jsou realizovány v programu v shaderech s názvy *voxelization.vs*, *voxelization.fs*. Detekce kolizí vrcholů testovaného objektu s objemovými daty voxelizovaného objektu je provedena v shaderech s názvem *collision.vs*, *collision.fs*.

Protože je potřeba pro dosažení správné voxelizace objektu spustit fragmentový shader i na fragmenty, které jsou zakryty, je potřeba vypnout hloubkový test. Aby došlo k zápisu aktuálně zpracovávaného fragmentu testovaného vrcholu či objemového elementu součásti na správnou pozici ve výsledné textuře, je potřeba nastavit, aby při stanovení pozice fragmentu v textuře byla použita interpolace pozice fragmentu hodnotou nejbližšího souseda. Pro účel voxelizace byla nastavena ortogonální projekce, protože bylo požadováno, aby byla získána informace o objemové reprezentaci bez ohledu na vzdálenost od kamery. Aby bylo možné uložit informaci o objemovém elementu, zda leží uvnitř či vně objektu, byla použita celočíselná ne znaménková textura, jejíž prvky jsou RGBA hodnoty. Požadavek na ne znaménkové celé číslo je z důvodu využití všech 32 bitů barevného kanálu. Při využití pouze jedné textury by bylo dosaženo rozlišení modelu v ose z 128 řezů. Omezení na 128 bitů je z důvodu, že texel je typu RGBA, tzn. má čtyři barevné kanály ($4 \cdot 32 = 128$). Pro získání většího rozlišení voxelizovaného objektu byla v této práci použita technika pole textur (z angl. texture array), pomocí níž došlo k propojení více celočíselných textur do jednoho pole textur. Pro voxelizaci objektů ve scéně bylo v této práci použito jako dostatečné rozlišení 1024^3, pro které je došlo k propojení 8 2D RGBA textur o rozlišení 128 jednotek jak v ose x tak v ose y. V metodě *alignLCSToGCS* třídy *Voxelization* je provedeno umístění lokálního souřadnicového systému součásti do požadované pozice pro provedení jeho voxelizace tak, aby součást byla zarovnána a umístěna v definovaném ořezovém prostoru. V metodě *reuse* třídy *Voxelization* je proveden samotný proces voxelizace objektu.

V metodě *recompute* třídy *Collision* dochází k iterování objekty scény tak, že dojde vždy k voxelizaci jednoho objektu scény a otestování ostatních objektů scény s ním, pokud již neexistuje pro danou dvojici objektů detekovaná kolize. Nejprve jsou otestovány extrémní vrcholy konvexní obálky testované součásti s objemovými daty voxelizovaného objektu, pokud neexistuje jejich kolize, pak dojde k otestování vrcholů definující povrch objektu testované součásti s objemovými daty voxelizovaného objektu. V metodě *afterTest* dochází k analyzování, zda existuje ve vytvořené 1D textuře detekovaná kolize příslušné dvojice objektů. Vytvoří se virtuální framebuffer, který je propojen s 1D texturou, ve které bude na pozici odpovídající indexu kolidujícího vrcholu binární 1. Vrchol testovaného objektu je reprezentován jako RGBA hodnota (c_t), která má nastavenou binární 1 na pozici, která odpovídá souřadnici z vrcholu. RGBA hodnota (c_v) odpovídají texelu voxelizovaného objektu, který

má stejnou hodnotou souřadnice x a y jako testovaný vrchol. Detekce kolizí se provádí použitím bitového součinu na hodnotu c_t a c_v. Pokud existuje ve výsledné $1D$ textuře alespoň jeden bit nenulový, pak má testovaná součást kontakt s voxelizovaným objektem.

3.3 Vytvoření grafu interakce

Struktura grafu je uložena v matici incidence a samotná matice incidence je reprezentována pomocí vektoru vektorů. Ve třídě *InteractionGraph* jsou uloženy pro součásti soustavy následující informace: seznam součástí, s nimiž má daná součást kontakt, typ jejich spojení a vzdálenost jejich os. Index součásti ve vektoru odpovídá indexu součásti ve vstupním souboru s definicí scény, které jsou číslovány od nuly. V metodě *classificAxes* se stanoví vzájemná poloha os rotace dvojice součástí, které mají spolu kontakt, podle velikosti skalárního součinu směrových vektorů os rotace. V metodě *determineJunction* se stanoví typ spojení dvojice součástí, které mají spolu kontakt, podle tvaru součástí a podle stanovené poloze jejich os rotace.

3.4 Vytvoření dynamické vizualizace soustavy mechanických součástí

Ve třídě *BFS* je realizován výpočet parametrů pohybu součásti soustavy. V metodě *bfsGraphIter* je realizován průchod grafem interakce, ve které dochází k volání funkce *verifyDirRotationalAxis*, ve které je realizováno stanovení správné orientace os rotace součástí soustavy vůči sobě. Ve funkci *isMoveFeasible* je realizováno stanovení, zda lze provést animaci soustavy, a pokud ano, pak jsou v této funkci vypočteny parametry animace součástí soustavy. Nejprve dojde k nalezení podgrafů grafu interakce, které je realizováno ve funkci *findPaths*. Pokud soustava obsahuje nepohyblivou součást, pak ve funkci *reassemblyMovement* se vypočte rychlost rotace součásti okolo nepohyblivé součásti a stanoví se, zda je taková animace možná. Vypočte se rychlost rotace součástí soustavy, které jsou obsaženy v aktuálně analyzovaném podgrafu grafu interakce (funkce *computeMovementSpread*). Po vypočtení parametru animace součásti se stanoví, pokud již jsou pro tuto součást nastaveny parametry pohybu, zda nedochází ke kolizi již vypočtených parametrů animace součásti s nově vypočtenými parametry (funkce *checkParameters*). Ve třídě *BFS* jsou uloženy podgrafy grafu interakce ve vektoru *paths*, které vzniknou, pokud budou nalezeny na cestě směrem od řídící součásti fixní součásti nebo součásti, které mají koaxiální osu s fixní součástí. Každý podgraf obsahuje list, jehož mohutnost je rovna počtu prvků v analyzované scéně. Pokud má součást kontakt s nějakou součástí ze seznamu, pak je obsaženo v tomto listu, jaký typ spojení mají tyto dvě součásti a jakou mají vzdálenost jejich rotační osy. Dále tento list obsahuje indexy nepohyblivých součástí v podgrafu, list indexů součástí, které mají koaxiální osu s osu rotace nepohyblivé součásti, index součásti, která byla zpracována v tomto podgrafu jako první a pro kterou jsou známy její parametry animace. Vypočtené informace o pohybu součásti jsou uloženy ve třídě *AnimationInfo*.

3.5 Vytvoření statické ilustrace soustavy součástí s grafickými informacemi

Ve třídě *Arrows* a *BoundingBox* této práce je umístěna logika problematiky umísťování šipek do statické ilustrace.

Nejprve byly seřazeny součásti směrem od kamery podle hodnoty souřadnice z po provedení jejich transformace modelovací, pohledovou a projekční maticí (funkce *sortCentresOfMass* třídy *Arrows*). Ve funkci *recompute* téže třídy jsou vytvořeny nejprve šipky pro pohyb součásti okolo nepohyblivé součásti a až poté dojde k vytvoření šipek pro rotaci okolo své rotační osy. Dále popisované funkce jsou obsaženy ve třídě *BoundingBox*. Vytvoření šipek pro rotaci součásti okolo nepohyblivé součásti je proveden voláním funkce *findArrowFixedMovement*. Nalezení pozice šipek pro rotaci součásti okolo nepohyblivé součásti je provedeno ve funkci *generatePositionsFixedMovement*. Vytvoření šipek pro rotaci součásti okolo vlastní osy rotace je vytvořeno zjištěním, jaké vzorky jsou zakryty (funkce *checkCoveredSamples*), poté bylo vytvořeno ohodnocení z hlediska silového modelu (funkce *evalutionAttractiveRepulsiveForce*), poté bylo zjištěno ohodnocení z hlediska slábnutí (funkce *preventArrowAccumulation*), poté bylo ve funkci *checkCoveredArrows* stanoveno, kolik fragmentů šipky je vidět a dle celkových nároků na ohodnocení byla nalezena nelepší šipka. Ve třídě *BoundingBox* je uložena informace o vzorku a jeho ohodnocení dle všech kritérií (poloměr, řez, celkové, slábnutí, síly), počtu viditelných fragmentů obou druhů šipek (vykreslovaných pro podstavu i plášť objektu), preferovaném druhu šipky a vykresleném výsledném druhu šipky, pozici vzorku, hodnot dle modelu slupek a vzdálenosti ve směru řezu i slupky. Ve funkci *sampleBB* je realizováno vytvoření vzorků uvnitř obálky součásti a jejich iniciální ohodnocení *evaluateSample*. Nalezení nejlepšího vzorku je realizováno v metodě *findBestSample*. Stanovení preferovaného typy šipky pro vzorek je provedeno ve funkci *determineArrowType*. Funkce zarovnání šipky na pozici odpovídající pozici vzorku a její zarovnání s lokálním souřadnicovým systémem součásti je realizována ve funkci *locationArrow*.

Kapitola 4

Testování

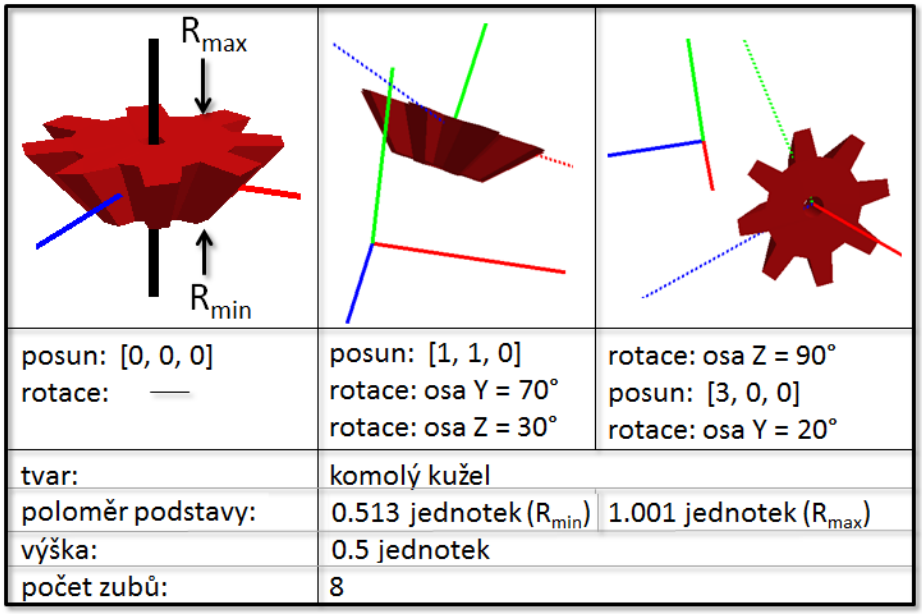
V této kapitole jsou nejprve uvedeny dosažené výsledky stanovení vlastností pro různé druhy součástí, které byly použity v testovacích soustavách. Poté jsou uvedeny vytvořené grafy interakce a vytvořená dynamická vizualizace a statická ilustrace se šipkami analyzovaných testovacích soustav. Výsledky dynamické vizualizace pohybu soustavy součástí jsou uvedeny v přiložených videích na CD. V příloze C jsou uvedeny další testovací soustavy, které se skládají z 10 až 30 součástí, jejich vytvořené grafy interakce a příklady vytvořených statických ilustrací jejich pohybu se šipkami.

4.1 Výsledky testování vlastností součásti

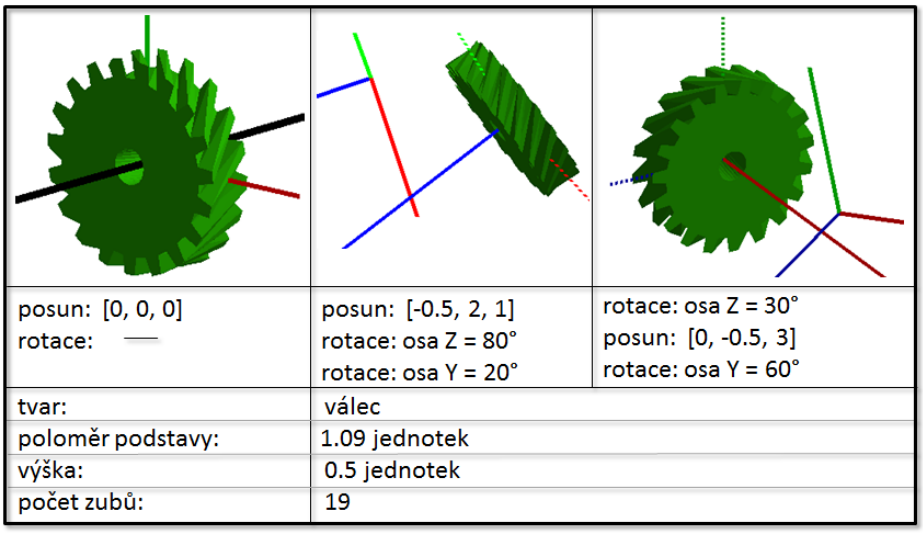
Pro otestování funkčnosti stanovení vlastností mechanických součástí (poloměr podstav, počet zubů, výška, tvar a rotační osa) byly umístěny součásti do různých poloh. Aplikované transformace, které změní polohu součásti, jsou uvedeny v tabulce v takovém pořadí, ve kterém byly na součást aplikovány. Transformace jsou prováděny vůči počátku globálního souřadnicového systému. Určené vlastnosti součásti byly zpětně zkontrolovány oproti údajům z modeláře. Na následujících obrázcích je globální souřadnicový systém znázorněn bázovými vektory (červená barva pro osu x, zelená barva pro osu y a modrá barva pro osu z) a je vykreslen plnými čarami. Lokální souřadnicový systém součásti má počátek v těžišti objektu a jeho bázové vektory (se stejnými barvami jako pro případ globálního souřadnicového systému), ale jsou vykresleny čárkovaně, pouze bázový vektor součásti, který je rotační osou součásti, je znázorněn plnou čarou. Rotační osa součásti je zobrazena v dále uvedených obrázcích vždy pro součást v základní poloze černou plnou čarou. Výsledky testování pro součást, která má tvar komolého kužele, jsou uvedeny v tabulce na obrázku 4.1. Pro tuto součást existují dvě podstavy, které mají odlišný poloměr.

Výsledky testování pro součást, která má tvar válce a je použita pro spojení typu spirálové ozubené soukolí, jsou uvedeny na obrázku 4.2. Tato součást má šikmé zuby, pro které není přesně definován jejich úhel, ale musí platit podmínka, že zuby dvojice tohoto tvaru součástí musí do sebe zapadat.

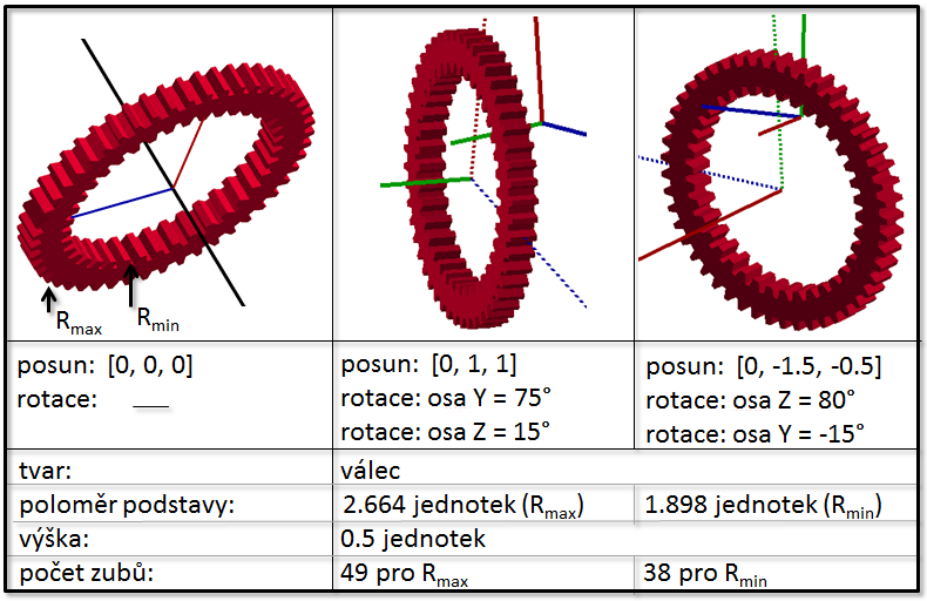
Na obrázku 4.3 jsou uvedeny stanovené vlastnosti ozubeného kola, které má vnitřní i vnější zuby a které může mít spojení se součástmi soustavy buď typu válec na válci nebo válec ve válci.



Obrázek 4.1: Vlastnosti ozubeného kužele, které může mít spojení se součástmi soustavy buď kuželové ozubené soukolí nebo osa. Tato součást má rovné zuby.



Obrázek 4.2: Vlastnosti ozubeného kola, které může mít spojení se součástmi soustavy buď spirálové ozubené soukolí nebo hřídel. Tato součást má šikmé zuby.



Obrázek 4.3: Vlastnosti ozubeného kola, které má vnitřní a vnější zuby a které může mít spojení se součástmi soustavy buď válec na válci nebo válec ve válci.

Pro neozubenou součást tvaru válce, která může mít pouze spojení se součástmi soustavy typu hřídel, jsou uvedeny její stanovené vlastnosti na obrázku 4.4.

Pro ozubené kolo, které nemá otvor na hřídel a může mít spojení se sousedními součástmi soustavy typu buď válec na válci nebo válec ve válci, jsou jeho stanovené vlastnosti zobrazeny na obrázku 4.5.

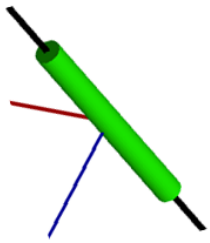
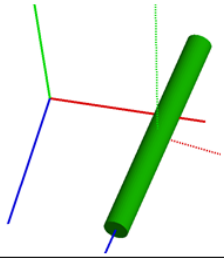
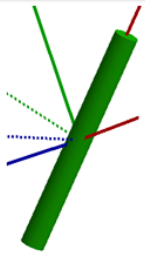
Pro ozubené kolo, které má otvor na hřídel, jsou možná spojení stejná jako pro ozubené kolo bez otvoru, navíc je možné spojení typu osa s hřídelí. Stanovené parametry tohoto typu součásti jsou zobrazeny na obrázku 4.6.

4.2 Výsledky testování grafu interakce součásti

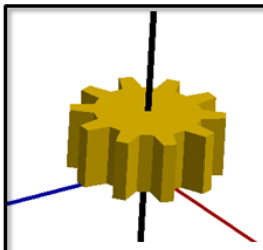
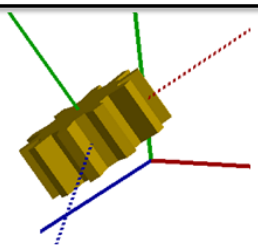
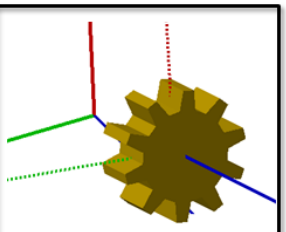
Nalezení kontaktu mezi dvojicí součástí soustavy signalizuje vytvořená hrana v grafu interakce mezi příslušnou dvojicí uzlů.

Pokud nejsou zadány typy součástí soustavy ze vstupního souboru, pak při spuštění aplikace je zobrazena každá součást soustavy, která není pomocnou součástí, po aplikování transformací pro zarovnání tohoto objektu do krychle, která bude voxelizována. V tomto kroku lze zkontrolovat, zda výchozí poloha objektu pro voxelizaci je správně stanovena. Pokud totiž není objekt celý umístěn uvnitř této krychle, tzn. nějaká jeho byť minimální část je mimo tuto krychli, pak voxelizace skončí nekorektně z důvodu použití bitové operace bitové nonekvivalence pro voxelizaci.

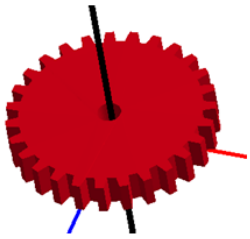
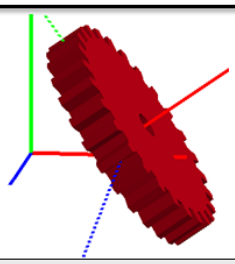
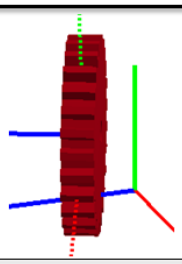
Z grafu interakce lze zjistit, zda byl správně stanoven typ spojení součástí podle označení, které je uvedeno pro každou hranu tohoto grafu. Popisek hrany grafu interakce může být

		
posun: [0, 0, 0] rotace: —	posun: [1.5, 0.2, 0] rotace: osa X = 90° rotace: osa Y = -5°	posun: [0, 0, 0] rotace: osa Y = 45° rotace: osa Z = 45°
tvar:	válec	
poloměr podstavy:	0.1574 jednotek	
výška:	3.0355 jednotek	
počet zubů:	—	

Obrázek 4.4: Vlastnosti neozubené součásti tvaru válce, která může mít spojení se součástmi soustavy pouze typu hřídel.

		
posun: [0, 0, 0] rotace: —	posun: [0, 0.3, -0.7] rotace: osa Z = -30° rotace: osa Y = 15°	posun: [0.05, -2, 0] rotace: osa X = -90° rotace: osa Y = 15° rotace: osa Z = -5°
tvar:	válec	
poloměr podstavy:	0.617 jednotek	
výška:	0.5 jednotek	
počet zubů:	10	

Obrázek 4.5: Vlastnosti ozubeného kola, které může mít spojení se součástmi soustavy buď typu válec ve válci nebo válec na válci. Tato součást má rovné zuby.

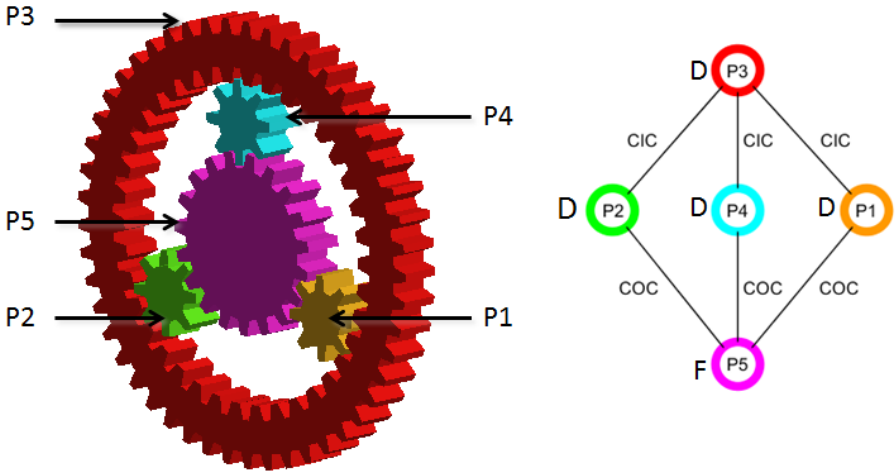
		
posun: [0, 0, 0] rotace: —	posun: [0.5, 1, -0.2] rotace: osa Z = 55°	posun: [0, -1, -1] rotace: osa X = -90° rotace: osa Y = -23° rotace: osa Z = 15°
tvar:	válec	
poloměr podstavy:	1.5095 jednotek	
výška:	0.5 jednotky	
počet zubů:	27	

Obrázek 4.6: Vlastnosti ozubeného kola, který má otvor pro hřídel, které může mít spojení se součástmi soustavy typu válec ve válci, válec na válci nebo osa. Tato součást má rovné zuby.

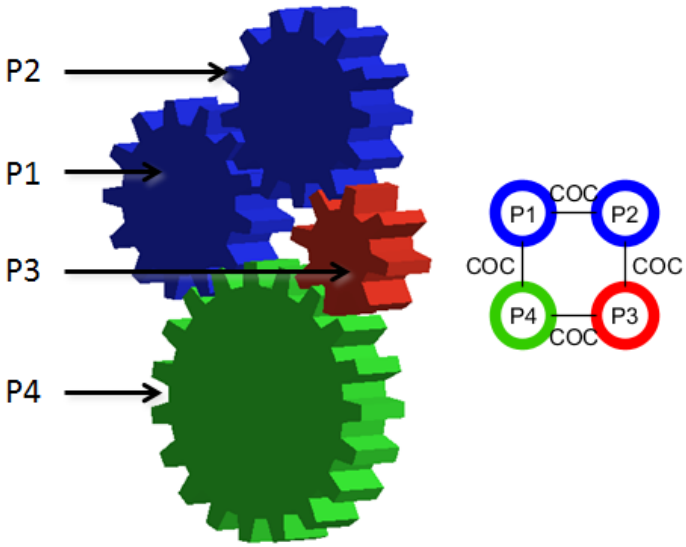
jednou z hodnot: CIC (válec ve válci), COC (válec na válci), A (hřídel), HG (spirálové ozubené soukolí) nebo BG (kuželové ozubené soukolí). Na obrázku 4.7, 4.8, 4.9, 4.10 a 4.11 jsou znázorněny příklady soustav mechanických součástí a jejich vytvořené grafy interakce. Barvy uzlů grafu interakce odpovídají barvám součástí, které představují. Do vytvořených obrázků, které znázorňují testovanou soustavu součástí, byly přidány šipky s popisky součástí soustavy, kde popisek součásti představuje její unikátní identifikátor, který je totožný s označením uzlu v grafu interakce. Popisky byly přidány z důvodu jednoznačného odlišení součástí soustavy i pro případ, kdy je v soustavě více stejně barevných součástí. Pro znázornění grafů interakce, které jsou uvedené v této práci, byl použit program *yEd* [35].

4.3 Výsledky testování statické ilustrace se šipkami

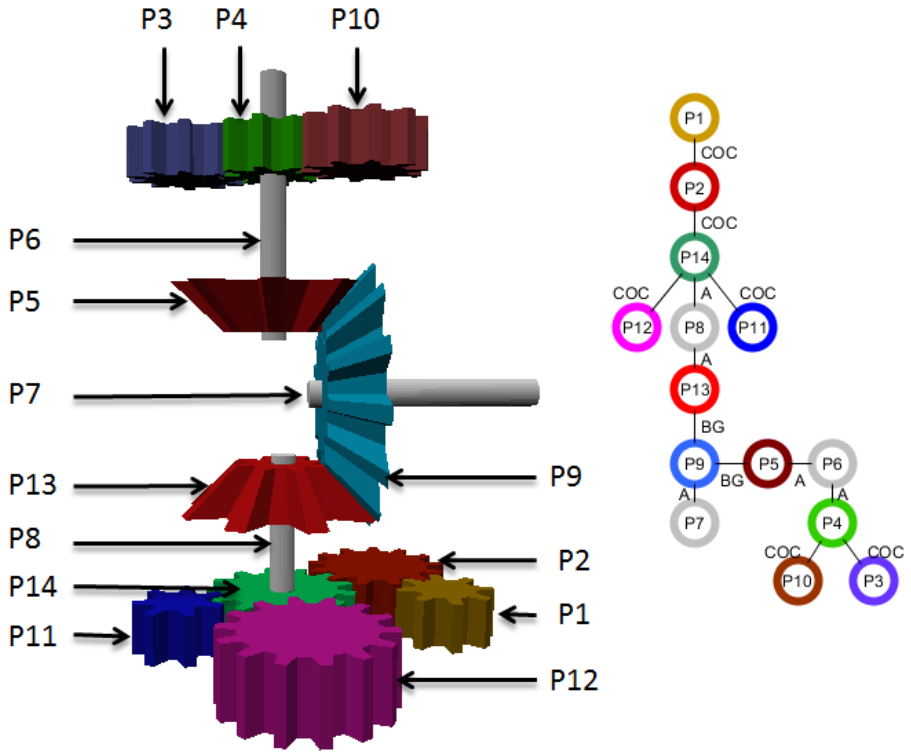
V této sekci jsou uvedeny výsledky statické ilustrace se šipkami pro pět testovaných soustav mechanických součástí. Pro jednotlivé soustavy jsou uvedeny hodnoty jejich vah (faktorů) ve výsledném ohodnocení vzorku. Zelené šipky ve vytvořených ilustracích přestavují rotaci součásti okolo rotační osy nepohyblivé součásti. Červené šipky přestavují rotaci součásti okolo své osy rotace. Černá součást ve vytvořené statické ilustraci znamená nepohyblivou součást a šedá součást představuje součást, pro níž neexistuje cesta mezi řídící součástí a touto součástí, na které by neležela nepohyblivá součást, a nebo přestavuje součást, která je málo viditelná ve scéně. Pro šedou ani černou součást nejsou generovány šipky. Poloprůhlednou barvou jsou znázorněny všechny pohyblivé součásti soustavy, které jsou dostatečně viditelné ve scéně a které nejsou řídící součástí. Řídící součást je znázorněna jako neprůhledná součást. Součást je dostatečně viditelná, pokud je její povrch nezakryt ostatními součástmi alespoň



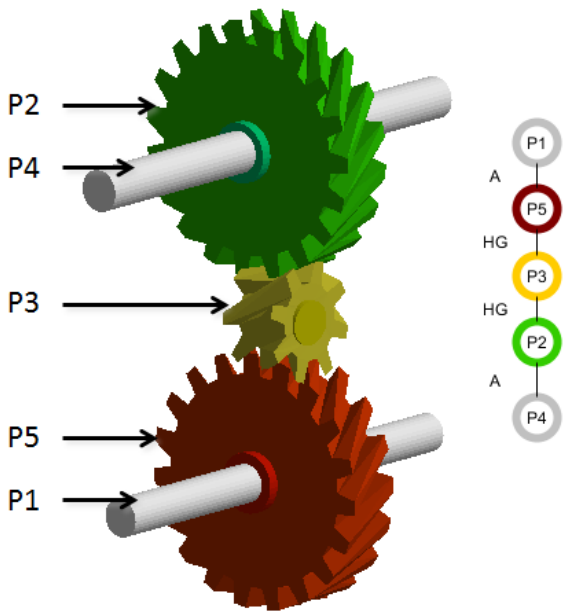
Obrázek 4.7: Testovací soustava 1 a její graf interakce.



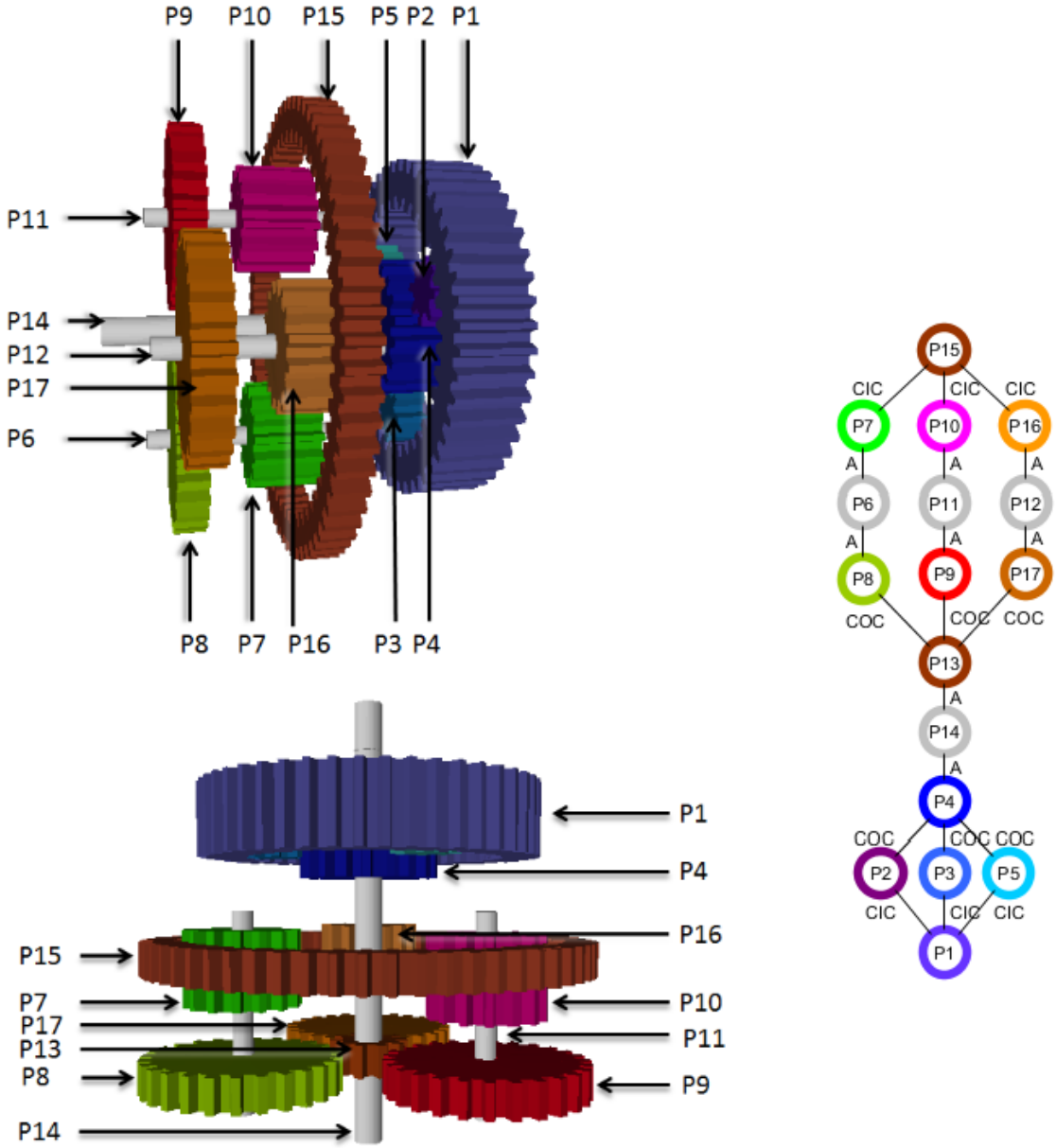
Obrázek 4.8: Testovací soustava 2 a její graf interakce.



Obrázek 4.9: Testovací soustava 3 a její graf interakce.



Obrázek 4.10: Testovací soustava 4 a její graf interakce.



Obrázek 4.11: Testovací soustava *test4* a její graf interakce. Typy spojení jsou BG (kuželové ozubené soukolí), HG (válcové ozubené soukolí), CIC (válec ve válci), COC (válec na válci) a A (osa).

z definovaného počtu procent. Pro dále popsané vytvořené ilustrace se šipkami soustavy součástí jsou v příslušných obrázcích uvedeny použité hodnoty faktorů, které udávají váhu příslušného modelu na výsledné ohodnocení vzorku a výběr nejlepšího umístění šipky.

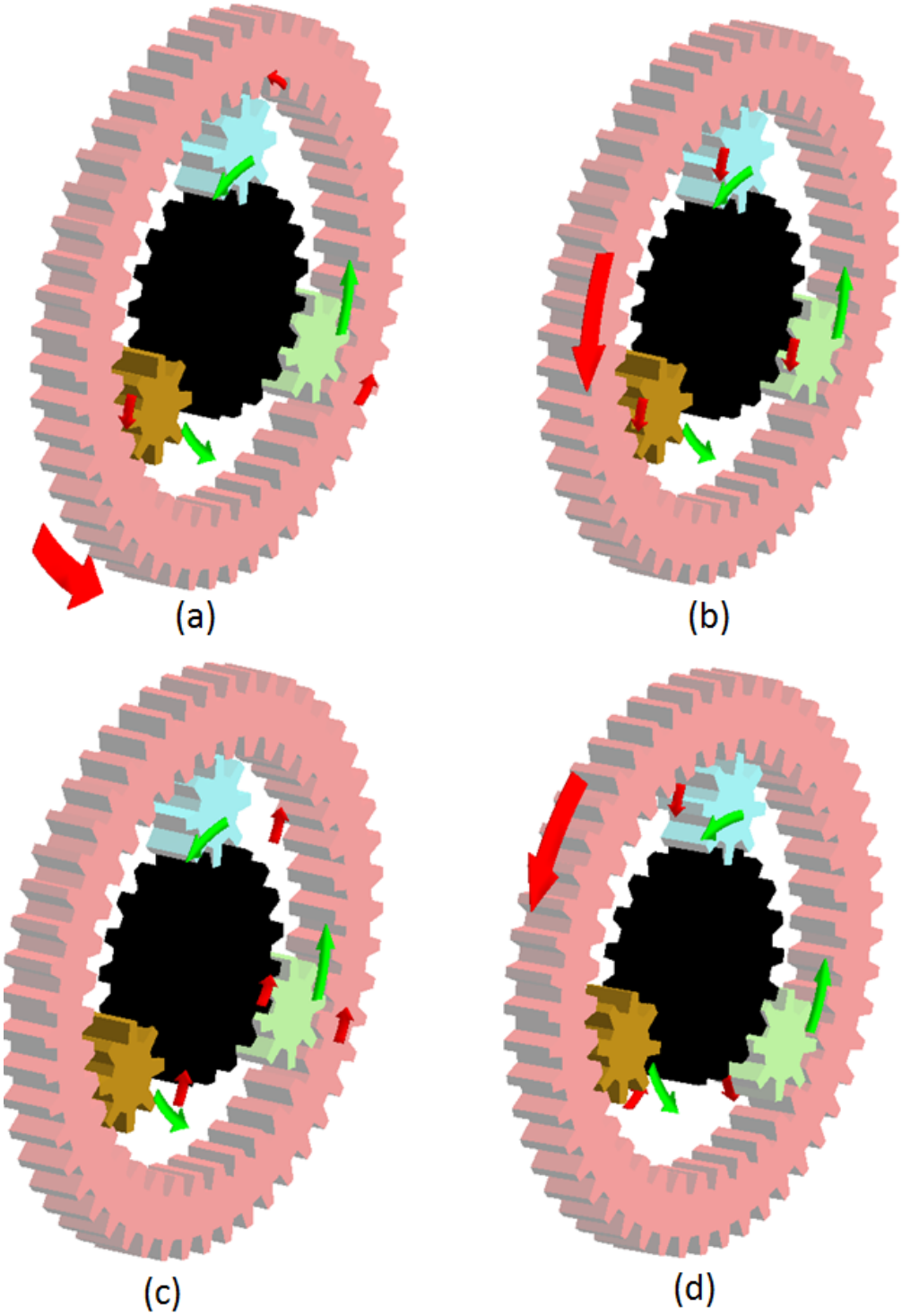
Na obrázku 4.12 je příklad soustavy, která se nazývá planetová převodovka, v níž se vyskytuje nepohyblivá součást (centrální kolo) okolo níž rotují planety (modré, žluté a zelené kolo) a vnější kolo (červené) má osu koaxiální s nepohyblivou součástí, proto není pro tuto součást vygenerována šipka pro její rotaci okolo nepohyblivé součásti.

Na obrázku 4.13 je uveden příklad druhé testovací soustavy, která je složena ze spojení součástí typu válec na válci (COC). Jedna ze čtveřice součástí je nepohyblivou součásti a zbylé součásti soustavy jsou nuceny rotovat okolo této součásti.

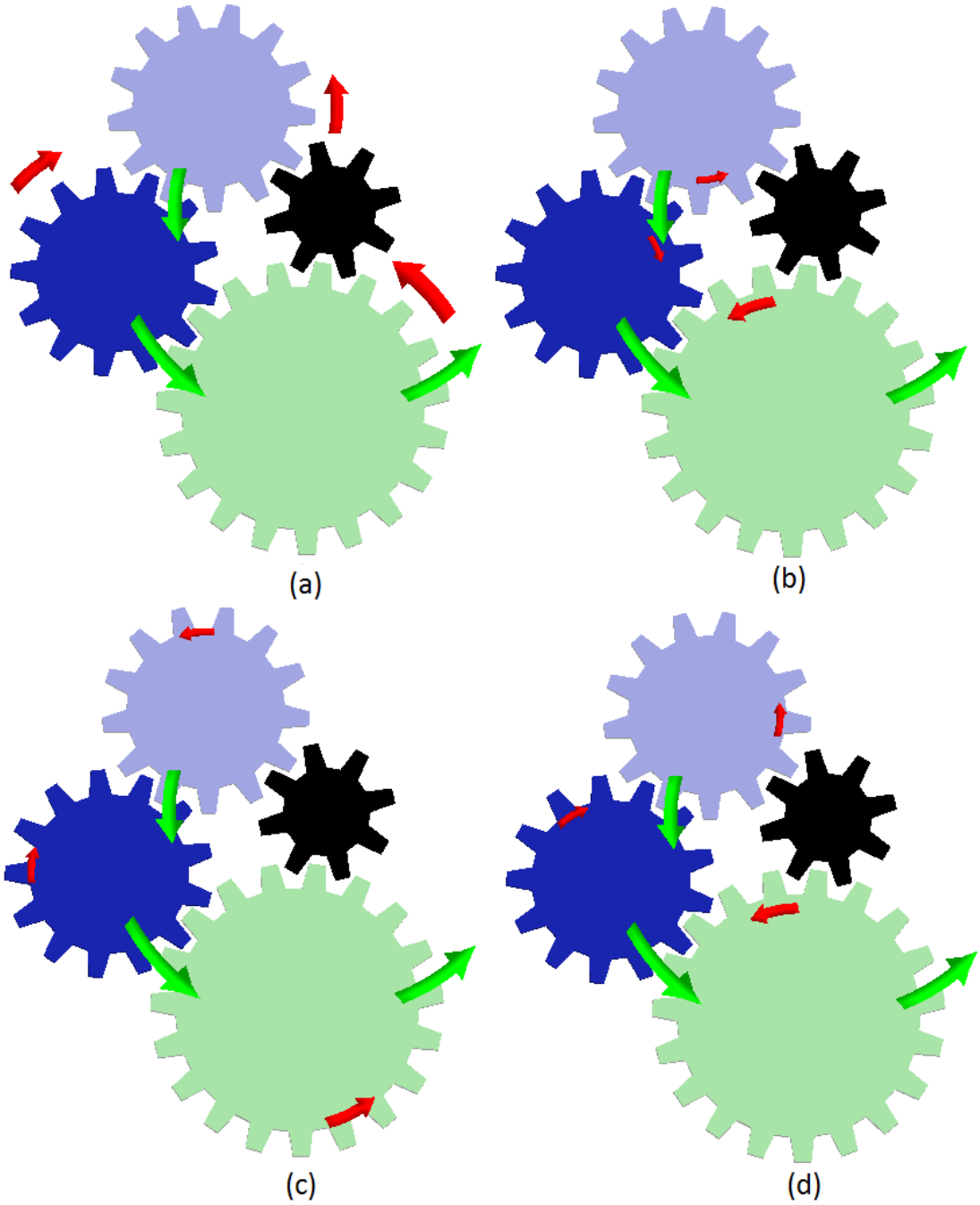
Soustava, která je uvedena na obrázku 4.15 je složena ze spojení součástí typu spirálové ozubené soukolí a hřídel.

Příklad složitější soustavy součástí je uveden na obrázku 4.14, která se skládá ze spojení součástí typu kuželové ozubené soukolí, osa a válec na válci. Pro tuto soustavu jsou nejprve uvedeny vygenerované statické ilustrace se znázorněným ohodnocení vzorků podle slupkového modelu a modelu řezů.

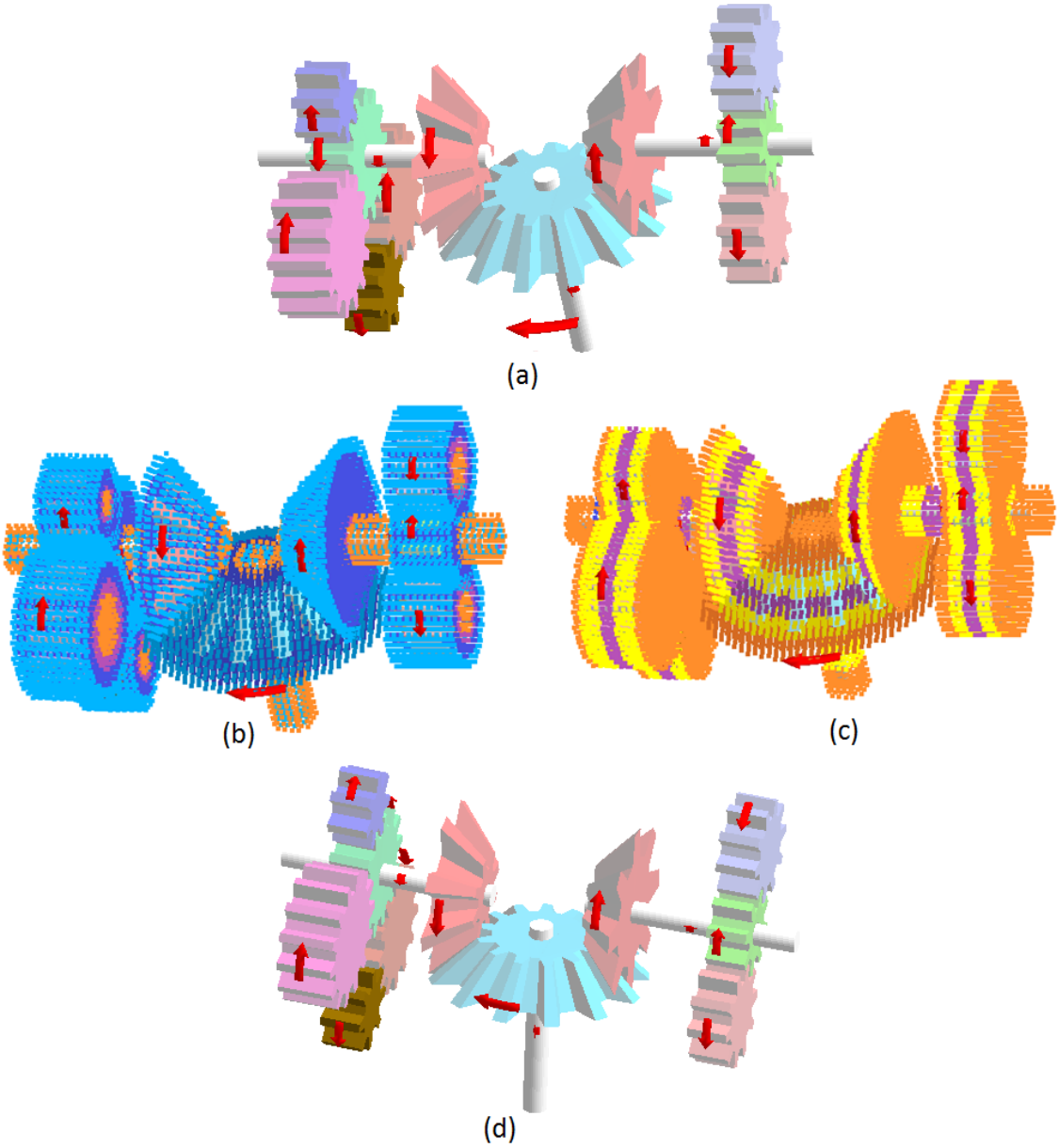
Na obrázku 4.16 jsou znázorněny příklady vytvořených statických ilustrací se šipkami pro soustavu součástí, která je složena ze spojení typu válec na válci, válec ve válci a hřídel. Příklad vytvoření statické ilustrace je uveden pro případ, kdy součást obsahuje a neobsahuje nepohyblivou součást.



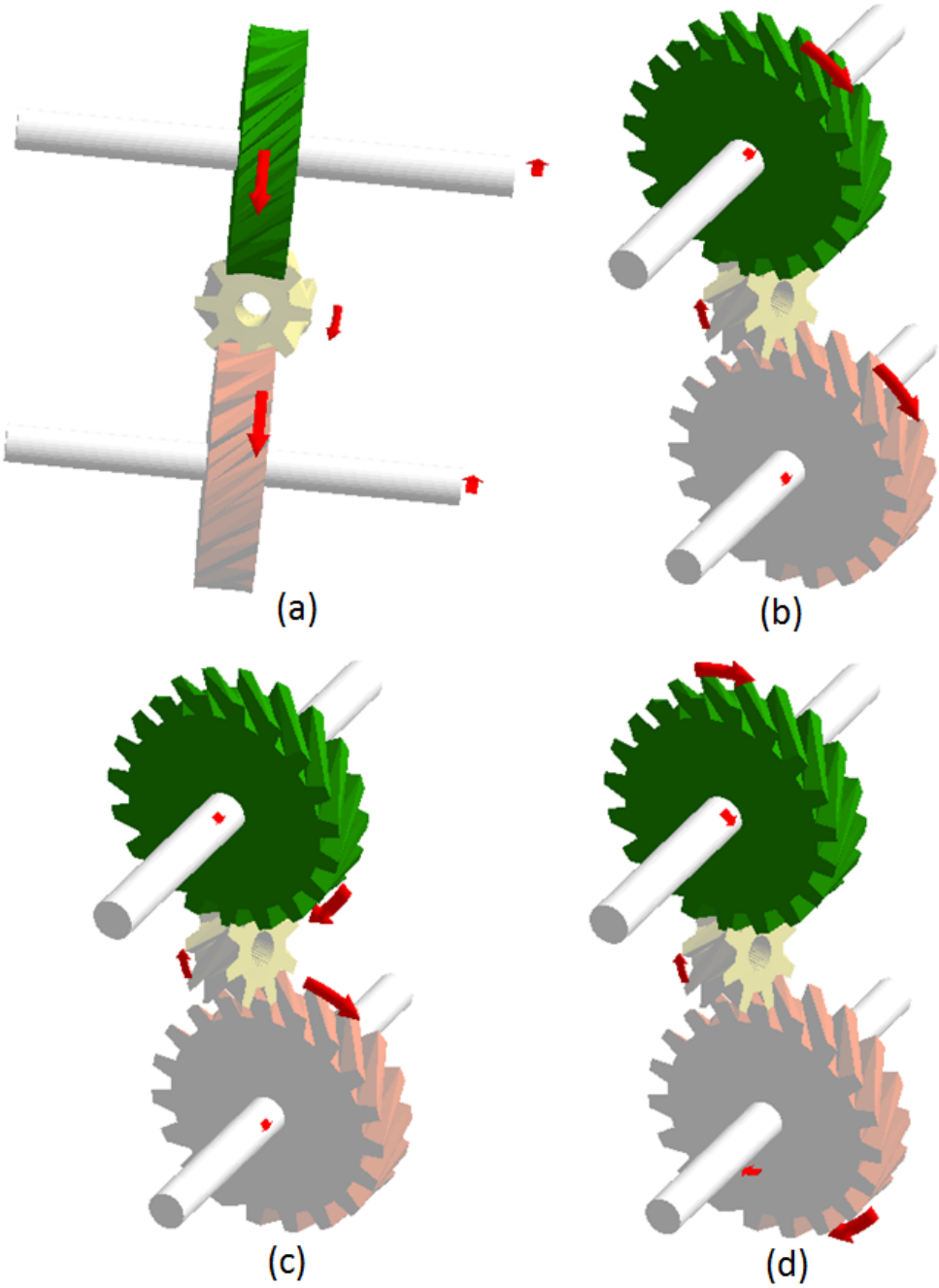
Obrázek 4.12: Statická ilustrace se šipkami pohybu soustavy s následujícími hodnotami faktorů: (a) $f_{position} = 0.5$, $f_{arrrow} = 0.1$, $f_{weak} = 1.0$; (b) $f_{position} = 0.5$, $f_{arrow} = 0.1$; (c) $f_{position} = 0.5$, $f_{arrow} = 0.1$, $f_{force} = 1.0$ (přitažlivé síly); (d) $f_{position} = 0.5$, $f_{arrow} = 0.3$, $f_{slice} = 1.0$, $f_{radius} = 1.0$



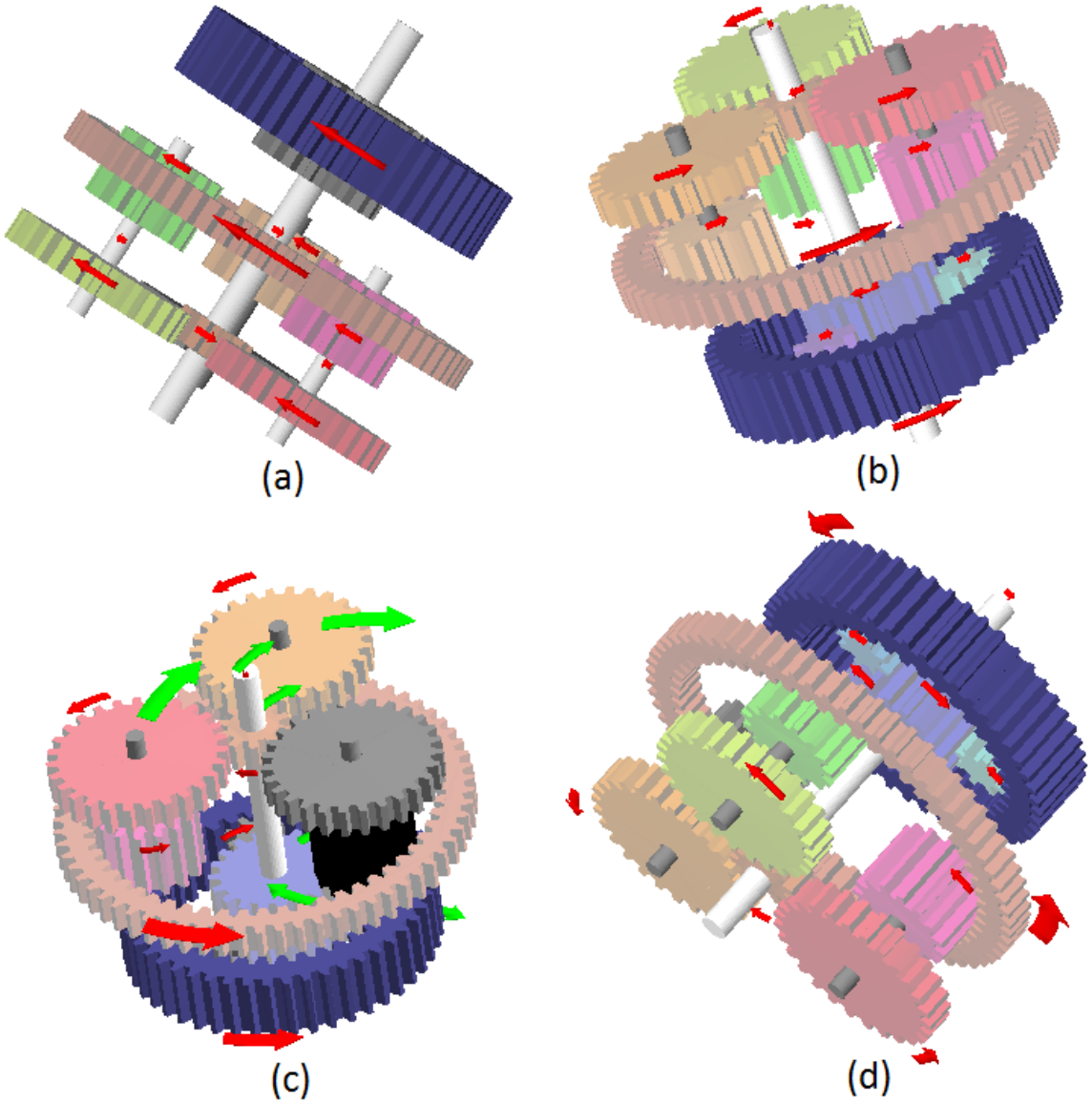
Obrázek 4.13: Statická ilustrace se šipkami pohybu soustavy s následujícími hodnotami faktorů: (a) $f_{position} = 0.5$, $f_{arrow} = 0.5$, $f_{radius} = 1.0$, $f_{force} = 1.0$ (přitažlivé síly); (b) $f_{position} = 0.5$, $f_{radius} = 1.0$, $f_{force} = 1.0$ (přitažlivé síly); (c) $f_{position} = 0.5$, $f_{radius} = 1.0$, $f_{weak} = 1.0$; (d) $f_{position} = 0.5$, $f_{radius} = 1.0$, $f_{weak} = 1.0$, $f_{force} = 1.0$ (přitažlivé síly)



Obrázek 4.14: Statická ilustrace se šipkami pohybu soustavy s následujícími hodnotami faktorů: (a) $f_{position} = 0.5$, $f_{radius} = 1.0$, $f_{slice} = 1.0$, $f_{arrow} = 0.5$; (b) ohodnocení vzorků pro (a) podle slupkového modelu; (c) ohodnocení vzorků pro (a) podle modelu řezů; (d) $f_{position} = 0.5$, $f_{radius} = 1.0$, $f_{slice} = 1.0$, f_{force} (odpudivá síla), $f_{arrow} = 0.1$



Obrázek 4.15: Statická ilustrace se šipkami pohybu soustavy s následujícími hodnotami faktorů: (a) $f_{arrow} = 1.0$; (b) $f_{radius} = 1.0$, $f_{slice} = 1.0$, $f_{arrow} = 0.2$; (c) $f_{radius} = 1.0$, $f_{slice} = 1.0$, $f_{arrow} = 0.2$, f_{force} (přitažlivé síly); (d) $f_{radius} = 1.0$, $f_{slice} = 1.0$, $f_{arrow} = 0.2$, $f_{force} = 1.0$ (odpudivé síly)



Obrázek 4.16: Statická ilustrace se šipkami pohybu soustavy s následujícími hodnotami faktorů: (a) $f_{position} = 0.5$, $f_{slice} = 1.0$, $f_{arrow} = 0.1$; (b) $f_{arrow} = 0.1$; (c) $f_{position} = 0.5$, $f_{arrow} = 0.5$, $f_{radius} = 1.0$; (d) $f_{position} = 0.5$, $f_{weak} = 1.0$, $f_{arrow} = 0.2$

Kapitola 5

Závěr

Cílem této diplomové práce bylo analyzovat existující metody pro ilustraci pohybu mechanických součástí a na základě analýzy navrhnout a implementovat aplikaci, která umožní vizualizovat pohyb soustavy součástí pomocí animace (dynamická vizualizace) a pomocí statického obrázku s grafickými informacemi v podobě šipek (statická ilustrace).

Pro vytvoření dynamické vizualizace soustavy mechanických součástí bylo potřeba zjistit, jak se bude chovat součást, jakmile jí bude udělena kinetická energie, a jak bude docházet k šíření kinetické energie soustavou směrem od řídící součásti. Tyto dvě informace byly spojeny do grafu interakce. Pro vytvoření statické ilustrace soustavy součástí bylo potřeba znát graf interakce soustavy a vypočtené parametry animace součástí soustavy.

Za účelem stanovení vlastností součástí soustavy byla analyzována technika pro nalezení osy symetrie objektu, nalezení rotační osy 3D objektu. Metoda realizovaná v této práci pro stanovení vlastností součásti byla založena na nalezení lokálního souřadnicového systému součásti, stanovení konvexní obálky součásti, nalezení jejích extrémních vrcholů a vytvoření shluků. Pomocí uvedených metod byla stanovena osa rotace součásti, poloměry podstav, počet zubů součástí, výška a tvar součásti.

Pro stanovení směru šíření kinetické energie soustavou bylo potřeba stanovit, jaké součásti soustavy spolu měly kontakt. Stanovení, jaké součásti spolu měly kontakt, bylo provedeno pomocí techniky detekce kolizí. Pro detekci kolizí součástí byly analyzovány metody SAT, BVH, Minkovského součet a detekce kolizí založená na voxelizaci. Detekce kolizí byla v této práci realizována na GPU s využitím bitového operátorů, celočíselných textur a voxelizace.

Aby bylo možné stanovit, jaké parametry animace mají součásti v soustavě, tak byla sloučena dohromady informace o součástech soustavy, které spolu mají kontakt, a informace o vlastnostech součástí a byl vytvořen graf interakce. Na základě průchodu grafem interakce došlo k vypočtení parametrů animace součástí soustavy a stanovení, zda je animace soustavy proveditelná.

Metody, které byly analyzovány, pro vytvoření statické ilustrace s pomocnými grafickými informacemi byly kauzální řetězec, klíčové snímky a šipky. Analyzovaná metoda pro umísťování šipek vytvořila šipky také pro spojení součástí. Realizovaná technika vytvoření statické ilustrace se šipkami v této práci vygenerovala vzorky ve zvětšené obálce součásti, na základě ohodnocení jednotlivých vzorků a viditelnosti šipky bylo stanoveno, jaká pozice byla nejvýhodnější pro vytvořený snímek soustavy. Pro soustavu součástí se vygenerovaly

šipky, které znázorňovaly rotaci součásti okolo vlastní osy rotace a osy rotace nepohyblivé součásti, pokud taková součást byla obsažena v soustavě. V této práce jsou také podporovány pomocné součásti, které sloužily jako kontext pro zobrazenou soustavu součástí.

Navržená aplikace byla implementována v jazyce C++. Implementovaná aplikace byla otestována na 13 modelech skládající se z 10 až 30 součástí. Pro všechny testované modely byly správně detekovány vlastnosti jednotlivých součástí i jednotlivé typy spojení. V případě chybných vstupních dat aplikace správně detekovala, že pohyb soustavy je neproveditelný. Výsledky testování ukazují, že je aplikace funkční. Aplikace má několik omezení, které budou nyní uvedeny. Tvar součásti musí být buď komolý kužel nebo válec. Součást nesmí obsahovat uvnitř svého objemu žádné přepážky a jeho povrch musí být uzavřený, aby došlo ke správné voxelizaci objektu. Pro vytvoření grafu interakce se nesmí měnit parametry animace součásti ani seznam součástí, s nimiž má součást kontakt. Řídící součástí může být pouze jedna součást soustavy, která nemůže být pomocnou součástí. Pokud soustava obsahuje nepohyblivou součást, znaménko rotace ramene je rovno +1 a poměr počtu zubů na analyzované cestě mezi řídící součástí a nepohyblivou součástí je roven 1, pak není proveditelná animace soustavy.

Přínosem této práce je vytvoření aplikace, která automaticky vygeneruje na základě načtených 3D objektů, které představují mechanické součásti soustavy, dynamickou vizualizaci a statickou ilustraci se šipkami, a která minimalizuje potřebné množství interakce uživatele.

V budoucnu je nutné se zaměřit na rozšíření vstupní množiny tvarů součástí a typů jejich spojení. Rozlišit i složitějších tvary objektů, jejichž tvar vznikne buď složením více základních tvarů a nebo deformací jednoho či více základních tvarů. Dále je nutné se zaměřit na takové součásti, pro které se mění množina součástí, s nimiž mají součásti kontakt, během provádění animace jejich pohybu. Přidat kritéria pro nalezení vhodnějšího umístění šipek (např. umístění šipky podle velikosti části (podstava a plášť) součásti, minimalizování počtu zakrytých fragmentů objektů šipkou, aplikování silového modelu a slábnoucího efektu na všechny viditelné fragmenty šipky a nejen na její střed). Dále je nutné se do budoucna zaměřit na urychlení výpočtu statické ilustrace součástí soustavy například realizováním techniky obálek pro vygenerované vzorky uvnitř zvětšených obálek součástí, které by urychlily výpočet ohodnocení například podle silového modelu či podle modelu slábnutí. V neposlední řadě by bylo potřeba vytvořit pokročilejší techniky statické ilustrace například s využitím kauzálního řetězce.

Literatura

[1] 3Dvia. Software –3dvia, September 2013. <http://www.3dvia.com>.

[2] AntTweakBar. Anttweakbar gui, November 2012. <http://anttweakbar.sourceforge.net/doc/>.

[3] Assimp. Assimp import library, November 2012. <http://assimp.sourceforge.net/>.

[4] Autodesk. Software – autocad, May 2013. <http://www.autodesk.cz/products/autodesk-autocad/overview>.

[5] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, December 1996.

[6] blender. Software –blender, September 2013. <http://www.blender.org>.

[7] Boost. boost c++ libraries, November 2012. <http://www.boost.org>.

[8] CamStudio. Camstudio, April 2013. <http://camstudio.org>.

[9] CGAL. Cgal mnohostěn, November 2012. <http://doc.cgal.org/latest/Polyhedron/classCGAL_1_1Polyhedron__3.html>.

[10] CGAL. Library – cgal, April 2013. <http://www.cgal.org>.

[11] CGAL. Library – cgal convex hulls 3d, April 2013. <http://www.cgal.org/Manual/latest/doc_html/cgal_manual/Convex_hull_3/Chapter_main.html>.

[12] Jung-Woo Chang, Wenping Wang, and Myung-Soo Kim. Efficient collision detection using a dual bounding volume hierarchy. In *Proceedings of the 5th international conference on Advances in geometric modeling and processing*, GMP'08, pages 143–154, Berlin, Heidelberg, 2008. Springer-Verlag.

[13] Leonaro da Vinci. Image – leonardo da vinci crossbow, July 2013. <http://ferociousstrength.com/mental-training-leonardo-da-vinci-horseshoe-bending/>.

[14] developer nvidia. Image – voxelization, May 2013. <http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter37.html>.

[15] Elmar Eisemann and Xavier Décoret. Single-pass gpu solid voxelization for real-time applications. In *Proceedings of graphics interface 2008*, GI '08, pages 73–80, Toronto, Ont., Canada, Canada, 2008. Canadian Information Processing Society.

[16] C. Ericson. *Real Time Collision Detection.* Real-time Collision Detection. Elsevier/Morgan Kaufmann Publishers, 2005.

[17] Efi Fogel and Dan Halperin. Exact and efficient construction of minkowski sums of convex polyhedra with applications. In *In Proc. 8th Workshop Alg. Eng. Exper. (Alenex06*, pages 3–15, 2006.

[18] GLEW. Glew, November 2012. <http://glew.sourceforge.net/>.

[19] GLM. Glm, November 2012. <http://glm.g-truc.net/0.9.4/index.html>.

[20] GLUT. Glut, November 2012. <http://www.opengl.org/resources/libraries/glut/>.

[21] S. Gottschalk, M. C. Lin, and D. Manocha. Obbtree: a hierarchical structure for rapid interference detection. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 171–180, New York, NY, USA, 1996. ACM.

[22] Christopher G. Healey and James T. Enns. Attention and visual memory in visualization and computer graphics, 2011.

[23] JAMA. Jama c++, November 2012. <http://math.nist.gov/tnt/download.html>.

[24] I. T. Jolliffe. *Principal component analysis.* Springer, New York, 1986.

[25] J. Milton. *Paradise lost, 1667.* Scolar P., 1968.

[26] N. J. Mitra, L. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (SIGGRAPH)*, 25(3):560–568, 2006.

[27] Niloy J. Mitra, Yong-Liang Yang, Dong-Ming Yan, Wilmot Li, and Maneesh Agrawala. Illustrating how mechanical assemblies work. *ACM Trans. Graph.*, 29:58:1–58:12, July 2010.

[28] Fakir S. Nooruddin and Greg Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9:191–205, 2003.

[29] P. Olšák. *Úvod do algebry, zejména lineární*, page 107. FEL ČVUT v Praze, 2007.

[30] OpenGL. Opengl, November 2012. <http://www.opengl.org/>.

[31] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, 1985.

[32] solidworks. Software – solidworks motion, May 2013. <http://help.solidworks.com>.

[33] G.J.A. van den Bergen. *Collision Detection in Interactive 3D Environments: Text*. Collision Detection in Interactive 3D Environments. Morgan Kaufmann Publishers, 2004.

[34] Richard Wright, Benjamin Lipchak, and Nicholas Haemel. *Opengl® superbible: comprehensive tutorial and reference, fourth edition*. Addison-Wesley Professional, fourth edition, 2007.

[35] yWorks. yed, March 2013. <http://www.yworks.com/en/products_yed_about.html>.

[36] Jiří Žára, Bedřich Beneš, Jiří Sochor, and Petr Felkel. *Moderní počítačová grafika*. Computer Press, Praha, 2. edition, 2005.

Příloha A

Instalační a uživatelská příručka

Uživatelská příručka aplikace realizované v této práci je přiložena na CD v adresáři *text*. V následujícím textu budou popsány instalační pokyny pro zprovoznění této práce.

V této práci byly použity následující knihovny:

- CGAL[10] (computational geometry algorithms library) verze 3.9,
- Boost[7] verze 1.47,
- AntTweakBar[2] verze 1.15,
- Assimp[3] (open asset import library) verze 3.0,
- JAMA[23] (numerical linear algebra library)verze 1.2,
- TNT[23] (template numerical toolkit) verze 1.2,
- OpenGL [30] (open graphics library),
- GLEW[18] (OpenGL extension wrangler library),
- GLM[19] (OpenGL mathematics),
- GLUT[20] (OpenGL utility toolkit).

Na CD je uložena spustitelná verze programu pro stroj, který splňuje následující uvedené požadavky:

- podpora rozšíření GL_EXT_gpu_shader4,
- minimální verze GLSL 3.3,
- OpenGL 4,
- podpora unsigned int 2D textur.

Knihovna CGAL představuje robustní knihovnu, která obsahuje algoritmy a datové struktury pro geometrické objekty se zaměřením na přesné algoritmy, které jsou asymptoticky rychlé. Knihovna CGAL byla použita pro výpočet konvexní obálky součásti. Knihovna CGAL potřebovala pro svoji funkčnost knihovnu Boost, která představuje množinu knihoven pro programovací jazyk C++, která obsahuje podporu pro struktury a operace pro lineární algebru, zpracování obrazu, regulární výrazy, atd. Knihovna AntTweakBar je C/C++ knihovna, která umožňuje vytvořit grafické uživatelské rozhraní do grafické aplikace, která je založena na OpenGL nebo DirectX. Pomocí této knihovny bylo vytvořeno ovládání aplikace, která byla vytvořena v této práci. Knihovna Assimp slouží pro načtení a zpracování geometrie scény, jejíž definice byla načtena ze souboru. Tato knihovna podporuje více různých formátů souboru, avšak pro účely této práce, bylo potřeba podporovat soubory typu OBJ, které jsou podporovány použitou knihovnou Assimp. Formát souboru OBJ představuje jednoduchý datový formát, který reprezentuje geometrii $3D$ objektů. Knihovna JAMA slouží pro řešení úloh z oblasti lineární algebry. Tuto knihovnu je možné použít jak pro programovací jazyk Java tak pro programovací jazyk C++. Pokud je použit programovací jazyk C++, pak používá tato knihovna knihovnu TNT (template numerical toolkit) pro operace na nízké úrovni. Protože byl realizován zdrojový kód této práce v programovacím jazyce C++, tak byla v této práci použita spolu s knihovnou JAMA, s jejíž pomocí byly nalezeny vlastní vektory a vlastní čísla pro vypočtenou kovarianční matici objektu, knihovna TNT. Knihovna OpenGL představuje standard pro multiplatformní rozhraní pro tvorbu aplikací počítačové grafiky. Knihovna GLEW je multiplatformní open-source C/C++ rozšíření OpenGL knihovny, které bylo v této práci potřeba podporovat z důvodu programování shaderů. Tato knihovna načte funkce ovladače OpenGL. Knihovna GLM představuje C++ matematickou knihovnu pro programování grafiky založenou na GLSL (OpenGL shading language). Tato knihovna byla použita pro snadnější a přehlednější práci s maticemi a vektory. Knihovna GLUT představuje rozhraní pro programování grafických aplikací založených na OpenGL.

Pro zkompilování aplikace je potřeba z uvedených knihoven, které jsou použity v této práci, nainstalovat knihovnu *CGAL* a *Boost*. Všechny knihovny v uvedených verzích, které jsou použity v této práci, jsou zabaleny a uloženy na přiloženém CD.

Vstupní a výstupní vrcholy $3D$ objektu byly uloženy do třídy Point_3 knihovny CGAL. Do této třídy byl přidán atribut *index*, který byl nastaven pro všechny vrcholy, které tvořily povrch objektu, a jehož hodnota odpovídala indexu do pole vrcholů, které tvořily povrch modelu, který byl číslován od nuly. Na základě tohoto indexu bylo známo při použití funkce convex_hull_3 knihovny CGAL, která pracovala s vrcholy typu Point_3, jaké vrcholy ze vstupní množiny vrcholů, které tvoří povrch objektu, tvoří vrcholy konvexního mnohostěnu. Z tohoto důvodu je potřeba změnit soubor, který je umístěn v knihovně CGAL: /include/CGAL/Cartesian/Point_3.h modifikovaným stejnojmenným souborem v této práci, který je přiložen na CD.

Příloha B

Konvexní obálky

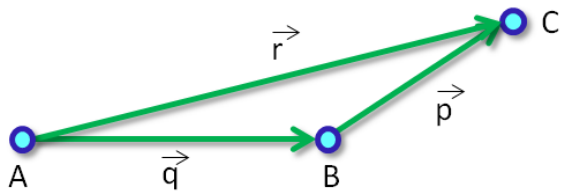
Preparata a kol. [31] rozdělují algoritmy pro stavbu konvexní obálky na offline, online a inkrementální algoritmy. Offline algoritmus potřebuje znát již na začátku úplnou množinu vstupních dat (např. selection sort). Online algoritmus postupně zpracovává vstupní data tak, jak přicházejí na vstup algoritmu (např. insertion sort). Inkrementální algoritmus zpracovává jeden vstupní element v jednom svém kroku. Existuje řada algoritmů pro stavbu konvexní obálky, které mají různou asymptotickou složitost. Vysvětlení základního principu stavby konvexní obálky je jednodušší provést pro $2D$ případ, proto si zde popíšeme některé algoritmy, které vytvoří konvexní obálku ve $2D$ a které jsou relativně jednoduše rozšířitelné do $3D$. Algoritmy pro tvorbu konvexní obálky, jejichž princip bude dále vysvětlen, budou:

- Jarvisův algoritmus,
- Quick hull,
- Rozděl a panuj.

Všechny tyto algoritmy jsou založeny na zjištění, zda leží bod vlevo, vpravo či na přímce ($2D$), rovině ($3D$). Situace je znázorněna na obrázku B.1. Určení umístění bodu C relativně vůči přímce $\vec{q}$ se provede na základě vypočtení determinantu dle vzorce B.1 a stanovení jeho znaménka (viz B.2). Popřípadě lze stanovit pozici bodu C relativně k přímce $\vec{q}$ výpočtem vektorového součinu směrových vektorů přímky $\vec{q}$, $\vec{r}$ a dle znaménka třetí souřadnice výsledného vektoru).

$$\det D = \begin{vmatrix} A_x & A_y & 1 \\ B_x & B_y & 1 \\ C_x & C_y & 1 \end{vmatrix} = (A_x - C_x) \cdot (B_y - C_y) - (A_y - C_y) \cdot (B_x - C_x) \tag{B.1}$$

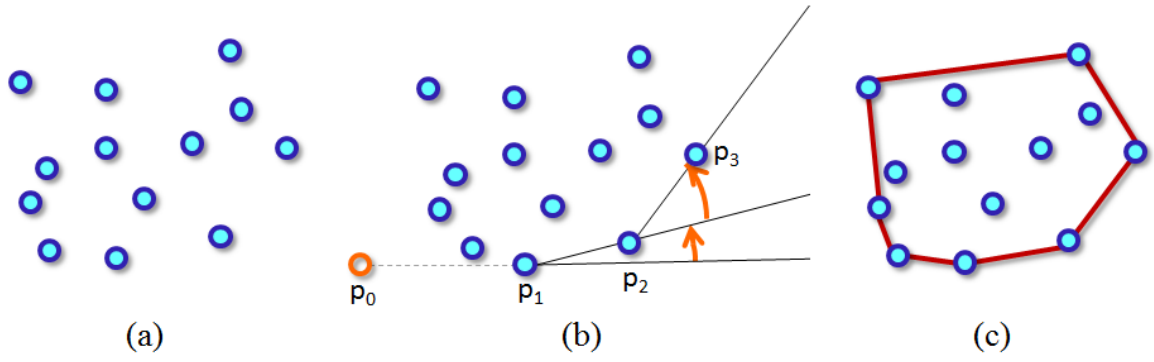
$$\text{bod C leží} = \begin{cases} \text{na přímce } \vec{q} & \text{pro sign(det D)} = 0 \\ \text{nalevo od přímky } \vec{q} & \text{pro sign(det D)} > 0 \\ \text{napravo od přímky } \vec{q} & \text{pro sign(det D)} < 0 \end{cases} \tag{B.2}$$



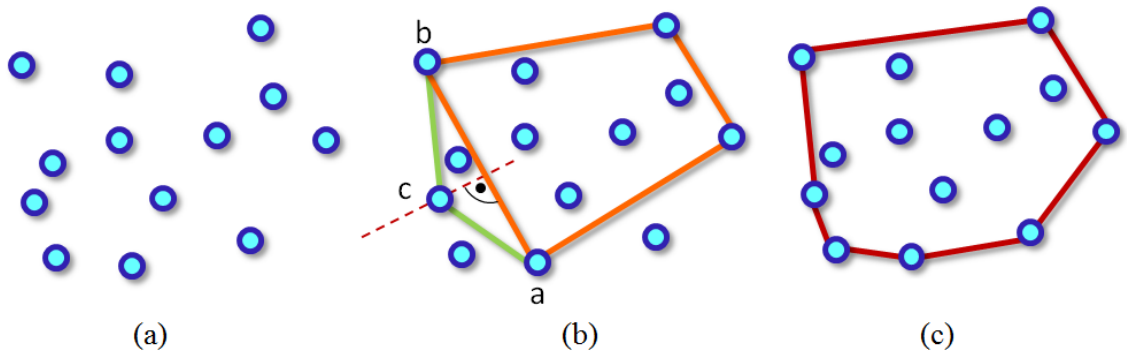
Obrázek B.1: Zjištění, zda bod C leží nalevo, napravo nebo na přímce $\vec{q}$.

B.1 Jarvisův algoritmus

Složitost algoritmu je rovna $n \cdot h$, kde n je mohutnost vstupní množiny dat a h je počet vstupních dat, které leží na konvexní obálce. Pokud budou všechna data ze vstupní množiny ležet na konvexní obálce, pak složitost algoritmu bude kvadratická. Je zřejmé, že složitost tohoto algoritmu je závislá na rozložení vstupní množiny dat. Tento algoritmus je vhodné použít pro případ, kdy bude ležet málo bodů ze vstupní množiny dat na konvexní obálce. Jedná se o obměnu řadícího algoritmu selection sort, který má asymptotickou složitost v nejhorším případě rovnu n^2. Princip Jarvisova algoritmu, který prezentují Preparata a kol. [31], je znázorněn na obrázku B.2. Řekněme, že budou přidávány body do seznamu bodů, které tvoří konvexní obálku, ve směru průchodu vrcholů konvexní obálky proti směru hodinových ručiček. Jako první bod konvexní obálky, která bude vytvořena nad vstupní množinou dat (viz obrázek B.2(a)), musí být přidán takový bod ze vstupní množiny, který je jejím extrémním bodem a který má tudíž např. nejmenší hodnotu y-ové souřadnice (označen na obrázku B.2(b) jako p_1). Pokud existuje více takových bodů, pak je potřeba vybrat takový bod z nalezených bodů, který bude mít např. minimální souřadnici x. Algoritmus hledá nejbližší bod k poslednímu doposud nalezenému bodu na konvexní obálce, který bude také ležet na této obálce, na základě minimalizování velikosti úhlu mezi dvojicí přímek, kde jedna přímka je vytvořena tak, že prochází doposud nalezenými posledními dvěma body konvexní obálky a druhá přímka prochází doposud nalezeným posledním bodem konvexní obálky a testovaným nejbližším bodem. Pokud však zatím byl nalezen pouze jeden bod konvexní obálky, pak je potřeba nalézt druhý bod, který bude také ležet na konvexní obálce. Z toho důvodu se vytvoří tzv. dočasný pomocný bod, který je označen na obrázku B.2(b) jako p_0 a který má souřadnice (-∞, $p1.y$). Nalezení nejmenšího úhlu je ekvivalentní nalezení takového bodu, kterým pokud vedeme přímku k bodu, který je posledním doposud nalezeným bodem na konvexní obálce, pak od této přímky leží všechny body vstupní množiny nalevo. Pro nalezení nejbližšího bodu k prozatím poslednímu nalezenému bodu p_i ze vstupní množiny dat, který bude ležet na konvexní obálce, se vezme libovolný bod p_j ze vstupní množiny dat a zkontroluje se, zda neleží nějaký bod ze vstupní množiny dat napravo od přímky, která bude vytvořena mezi bodem p_i, p_j. Pokud je splněna podmínka, pak tento bod p_j je přidán do seznamu vrcholů, které leží na konvexní obálce. Hledání vrcholů, které tvoří konvexní obálku, pokračuje do doby, nežli nalezený bod p_j není roven prvnímu bodu konvexní obálky tj. bodu p_1. Pokud však leží napravo od vytvořené přímky, na které leží body p_i, p_j, alespoň jeden bod ze vstupní množiny dat, pak se vybere libovolný bod, který leží napravo od této přímky, a je prohlášen za p_j. Tento postup nalezení bodu p_j se opakuje do té doby, než nebude existovat žádný bod napravo do testované přímky, na které leží body p_i, p_j.



Obrázek B.2: Jarvisův algoritmus: (a) Vstupní množina bodů. (b) Vytvoření pomocného bodu p_0 a stanovení bodů, které budou ležet na konvexní obálce. (c) Výsledná konvexní obálka vstupní množiny bodů.



Obrázek B.3: Quick hull: (a) Vstupní množina bodů. (b) Analyzování bodů, které leží vně čtyřúhelníku a které leží na jedné straně od analyzované úsečky *ab* čtyřúhelníka. Body uvnitř čtyřúhelníku budou odstraněny. (c) Výsledná konvexní obálka vstupní množiny bodů.

B.2 Quick hull

Asymptotická složitost tohoto algoritmu je O($n \cdot log(n)$) a v nejhorším případě n^2. Jedná se o obměnu řadícího algoritmu quick sort. Popisovaný princip metody je zobrazen na obrázku B.3. Tato metoda, jejíž princip prezentují Preparata a kol. [31], vychází z předpokladu, že při praktickém použití této metody bude většina bodů uvnitř konvexní obálky. Z tohoto důvodu se snaží metoda nalézt extrémní body, které budou součástí konvexní obálky. Ve $2D$ tvoří tyto extrémní vrcholy čtyřúhelník. Body, které jsou uvnitř tohoto čtyřúhelníku budou odstraněny. Postupně budou zpracovány všechny čtyři úsečky, které tvoří nalezený čtyřúhelník. Ze vstupní množiny zbývajících bodů, které leží nalevo od úsečky ab, bude nalezen takový bod c, který má největší kolmou vzdálenost od této úsečky. Odstraní se ze vstupní množiny bodů body, které leží uvnitř vytvořeného trojúhelníka abc. Rozdělí se vstupní množina na body, které leží nalevo od úsečky ac a které leží nalevo od úsečky bc. Stejný postup se aplikuje na rozdělené vstupní podmnožiny bodů. Nakonec se uloží hrana, pro kterou neexistuje žádný bod ze vstupní množiny, který by ležel nalevo od ní, do výsledné posloupnosti hran, které tvoří konvexní obálku.

B.3 Rozděl a panuj

Tato metoda je rozšířením řadícího algoritmu merge sort. Princip metody rozděl a panuj, kterou prezentují Preparata a kol. [31], je znázorněn na obrázku B.4. Fáze předzpracování vstupních dat, která je provedena během běhu algoritmu pouze při jeho inicializaci, seřadí body vstupní množiny dat podle hodnoty jejich x–ové souřadnice. Poté je rekurzivně rozdělena množina seřazených vstupních bodů vždy do dvou podmnožin. Toto rekurzivní rozdělování je prováděno do té doby, dokud není splněna ukončující podmínka pro rekurzivní dělení, kterou je maximální mohutnost podmnožiny. Nad dále nedělitelnou podmnožinou vrcholů se vytvoří konvexní obálka metodou hrubé síly. Poté dojde k postupnému slučování vzájemně se nepřekrývajících obálek (viz obrázek B.4(b) žluté úsečky) pro sousední podmnožiny bodů tak, že se nalezne jejich horní a dolní tangenta (viz obrázek B.4(b) zelené úsečky a podrobnější popis nalezení horní tangenty viz algoritmus 6), které spojí dvojici obálek. Spojením dvojice obálek vytvořenými tangentami se vyskytnou některé body, které jsou mezi těmito tangentami, uvnitř obálky, a proto tyto body musí být odstraněny. Uvedený princip je popsán algoritmem 5. Fáze předzpracování má složitost $n \cdot log(n)$, nalezení horní a dolní obálky má nejhůře lineární složitost, samotné sloučení dvou obálek má konstantní složitost a odstranění bodů ležící mezi přidanými tangentami má nejhůře lineární složitost. Celková asymptotická složitost tohoto algoritmu je tedy nejhůře $n \cdot log(n)$.

B.4 Quick hull ve 3D

Při stavbě konvexní obálky pomocí quick hull algoritmu, který prezentují Bradford a kol. [5], se nejprve vytvoří nadrovina. Pro eukleidovský prostor platí, že nadrovina rozdělí prostor na dva poloprostory. Pomocí znaménka vzdálenosti bodu k nadrovině se stanoví, zda bod leží nad či pod nadrovinou. Ke každé stěně konvexní obálky se přiřadí seznam bodů, které leží nad danou nadrovinou. Quick hull vybere nedegenerovanou množinu bodů pro vytvoření

Algoritmus 5 Vytvoř obálku(body P) [31]

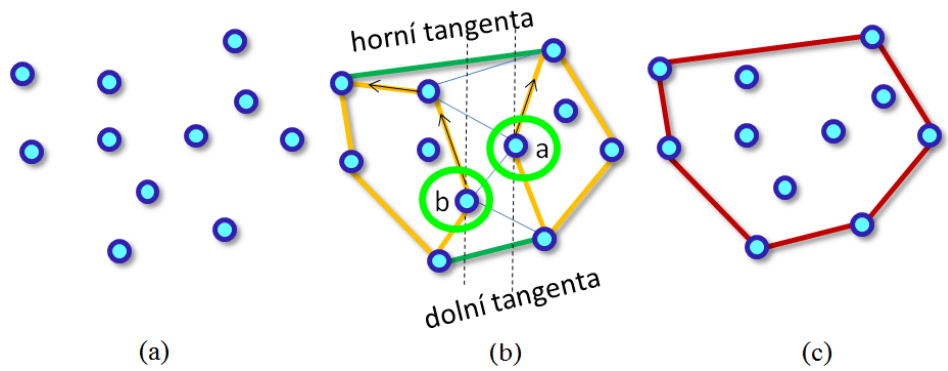
Vstup: seřazená množina bodů P podle x–ové souřadnice
Výstup: posloupnost vrcholů, které leží na konvexní obálce (CH)

```
if |P| ≤ 3 then
    počítej CH hrubou silou
    return CH
else
    Rozděl P do dvou podmnožin (L,R) podle x–ové souřadnice.
    Rekurzivně vypočítej H_L = Vytvoř obálku(L), H_R = Vytvoř obálku(R).
    CH = sluč(H_L, H_R) vypočítáním
    Horní tangenta(H_L, H_R),
    Dolní tangenta(H_L, H_R),
    odstraň body mezi těmito dvěma nalezenými tangentami.
    return CH
end if
```

Algoritmus 6 Horní tangenta(H_L, H_R) [31]

Vstup: dvě nepřekrývající se konvexní obálky
Výstup: horní tangenta sloučených vstupních konvexních obálek

```
a = bod, který má nejmenší x–ovou souřadnici pro konvexní obálku H_R
b = bod, který má největší x–ovou souřadnici pro konvexní obálku H_L
while ab není horní tangentou obálek H_L, H_R do
    while ab není horní tangenta obálky H_L do
        a = get následovníka a (směr CCW od a)
    end while
    while ab není horní tangenta obálky H_R do
        b = get předchůdce b (směr CW od b)
    end while
end while
return ab
```



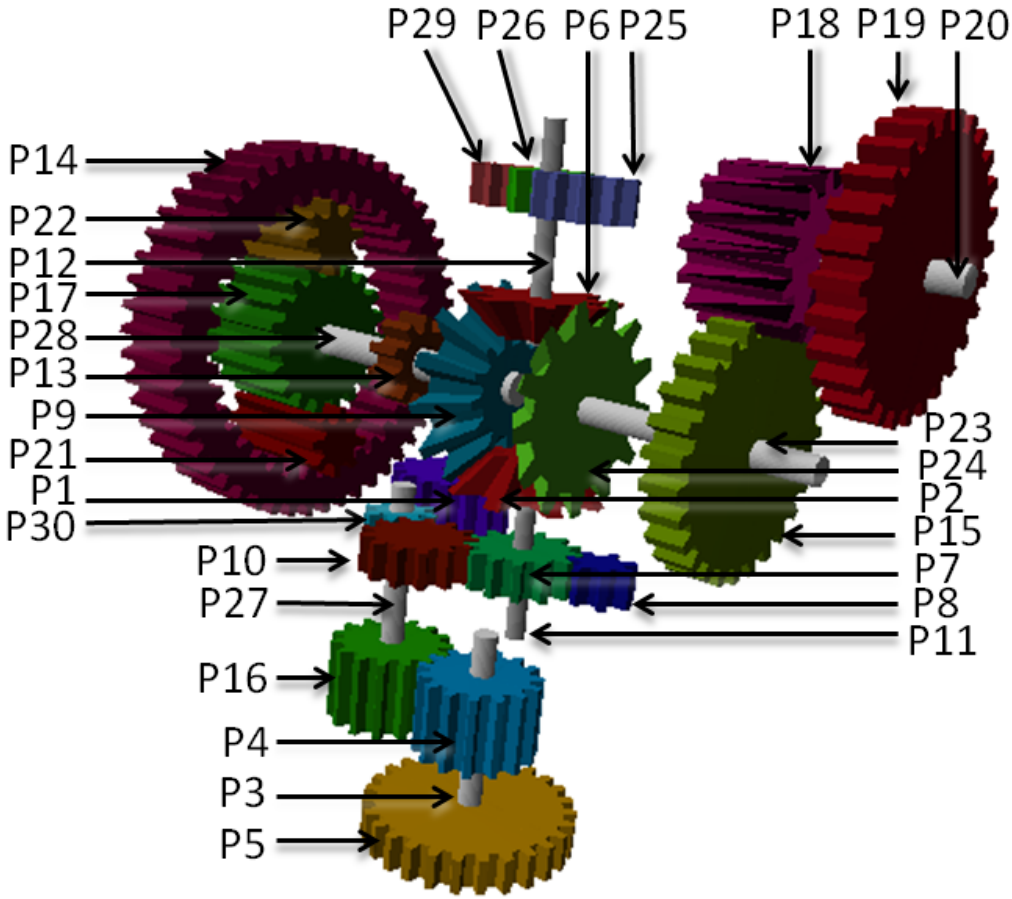
Obrázek B.4: Rozděl a panuj: (a) Vstupní množina bodů. (b) Vytvoření výsledné konvexní obálky sloučením dvou obálek pomocí nalezení horní a dolní tangenty. (c) Výsledná konvexní obálka vstupní množiny bodů.

iniciálního simplexu (obálky). K–simplexem je podle Preparaty a kol. [31] konvexní obálka $k+1$ afinních nezávislých bodů. Afinní nezávislé body jsou takové body, kde jeden bod může být vyjádřen jako afinní kombinace ostatních bodů. Afinní kombinací rozumíme takovou lineární kombinaci bodů, jejichž koeficienty se sčítají do 1 a jejichž koeficienty jsou reálnými čísly. D–polytopem je podle Preparaty a kol. [31] konvexní obálka konečné množiny bodů v eukleidovském prostoru dimenze D (E^D). Simplexem je podle Preparaty a kol. [31] takový polytop, jehož všechny body leží na konvexní obálce. Vybere se z množiny bodů, které leží nad nadrovinou, takový bod, který je nejvíce vzdálený od nadroviny. Vytvoří se stěny, které spojují nejvzdálenější bod s existující obálkou a stanoví se, jaké body leží nad nově vytvořenou obálkou. Stěny tvořící obálku obsahují informaci o sousedních stěnách. Při nalezení nové stěny je potřeba stanovit zdali bude již existující stěna obálky smazána a nahrazena nově vzniklou stěnou.

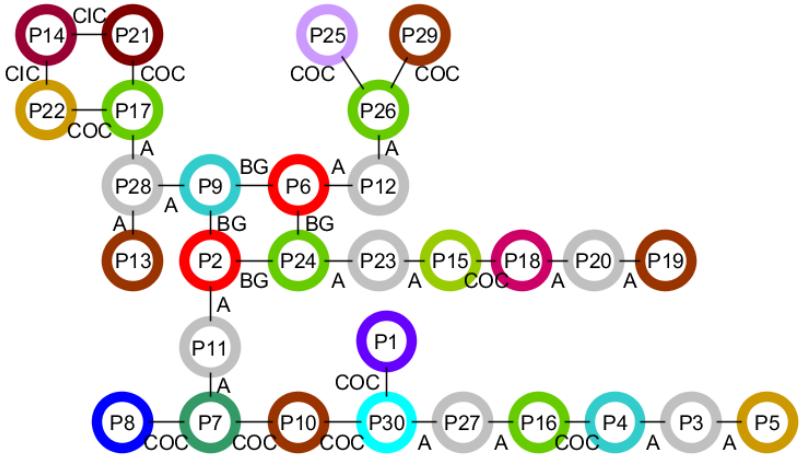
Příloha C

Výsledky testování

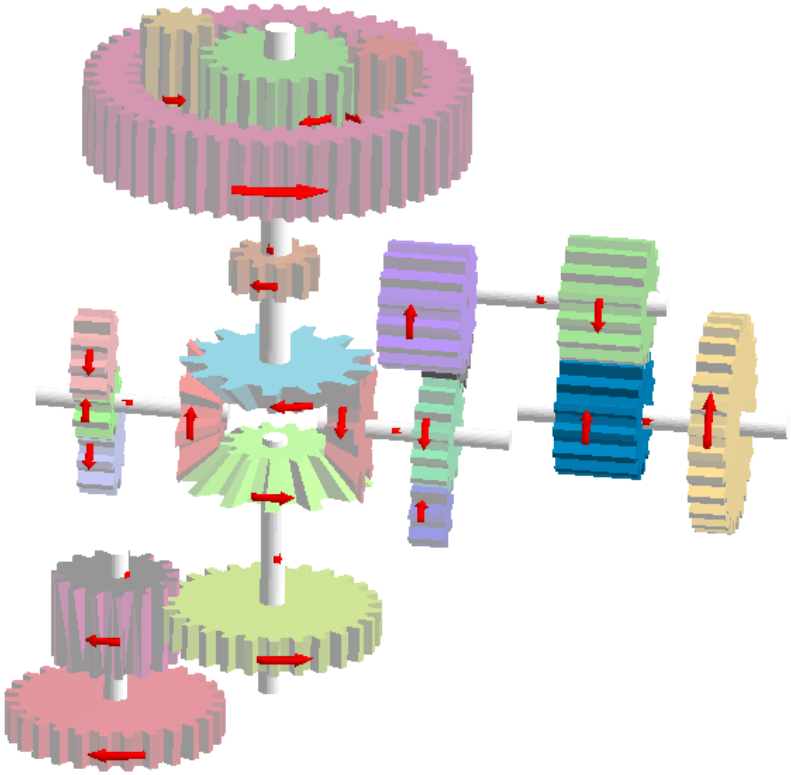
V této příloze jsou dále uvedeny soustavy součástí, které jsou tvořeny 10 až 30 součástmi, jejich grafy interakce a příklady jejich vytvořených statických ilustrací s uvedenými parametry pro jejich vytvoření.



Obrázek C.1: Testovací soustava *test6*.

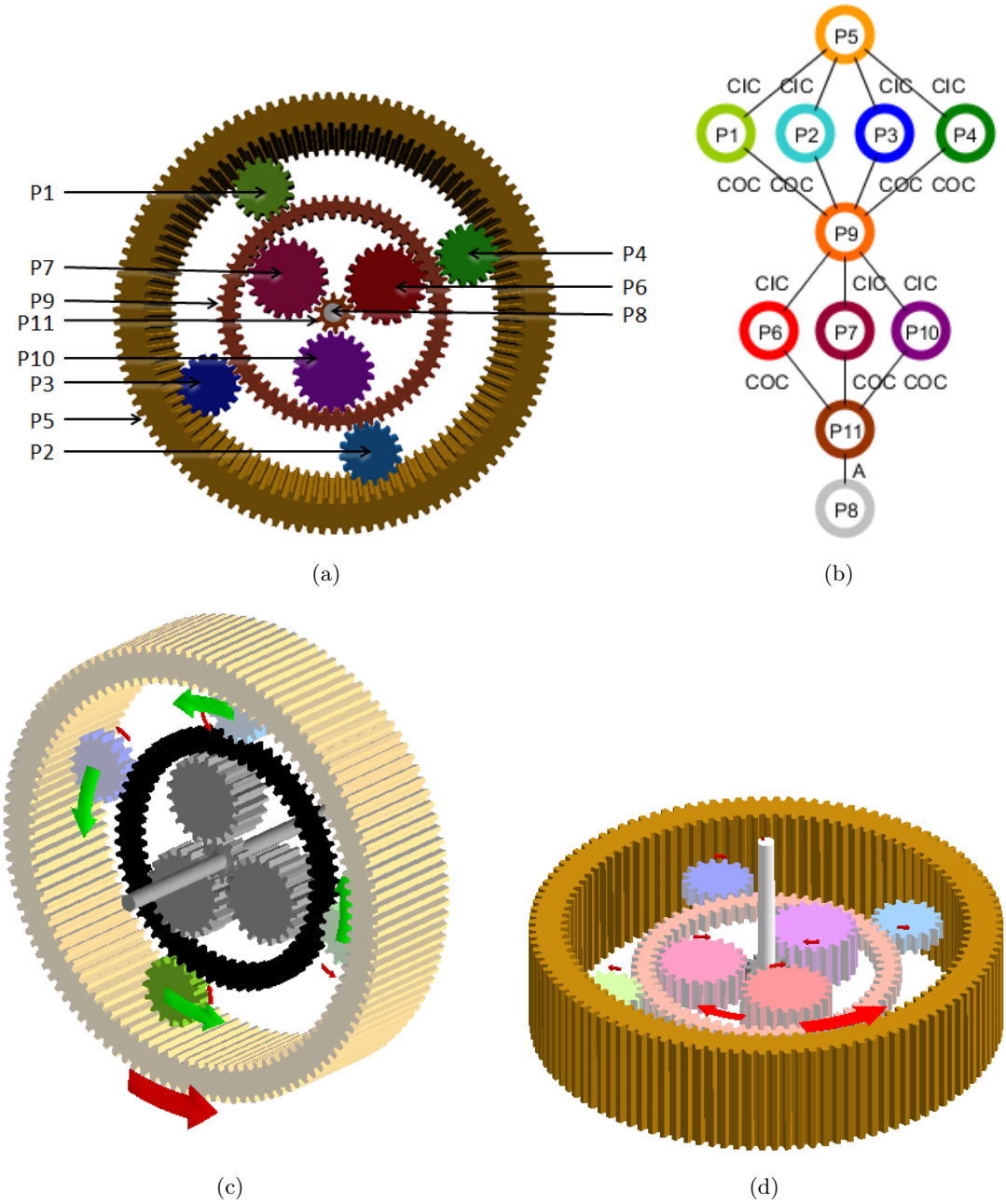


(a)

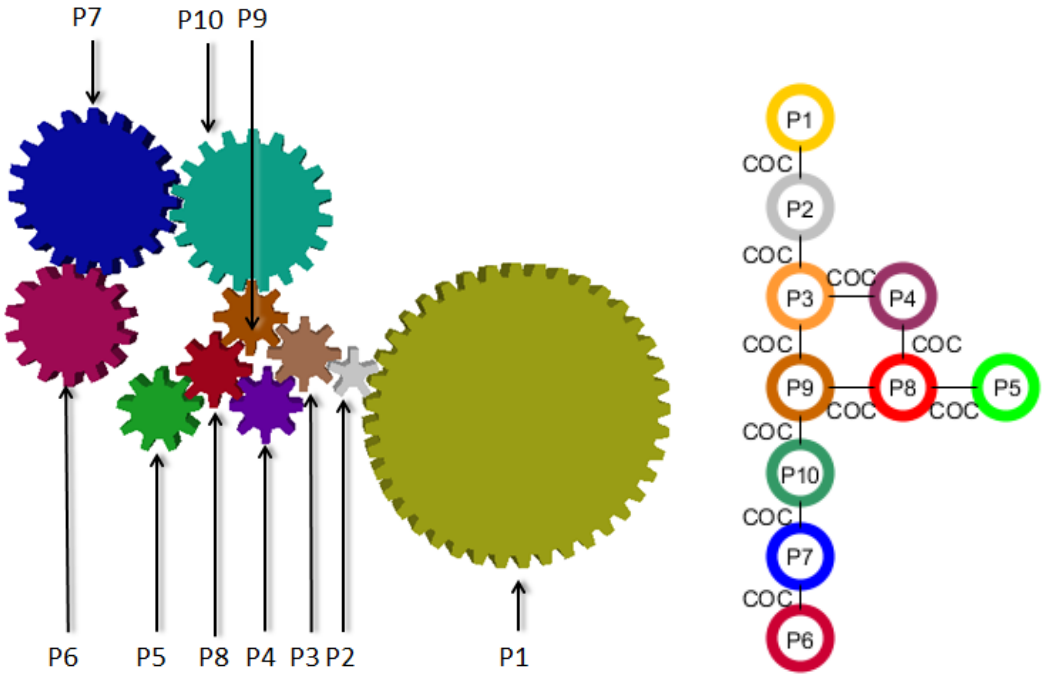


(b)

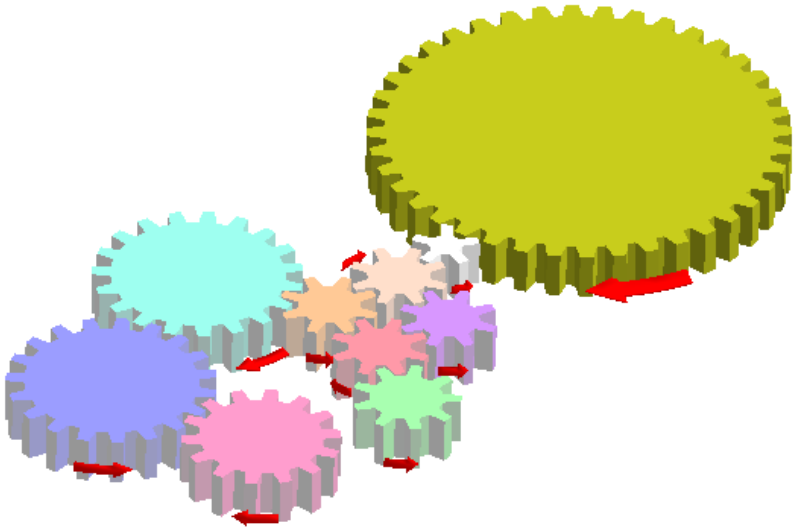
Obrázek C.2: (a) Graf interakce testovací soustavy *test6*. Typy spojení jsou BG (kuželové ozubené soukolí), CIC (válec ve válci), COC (válec na válci) a A (osa). (b) Příklad vytvořené statické ilustrace se šipkami testovací soustavy *test6*, která byla vytvořena s hodnotami faktorů: $f_{position} = 0.5$, $f_{radius} = 1.0$, $f_{slice} = 1.0$, $f_{arrow} = 0.3$.



Obrázek C.3: (a) Testovací soustava *test3* a její graf interakce (b). Typy spojení jsou CIC (válec ve válci), COC (válec na válci) a A (osa). Příklady statických ilustrací se šipkami, které byly vytvořeny s následujícími hodnotami faktorů: (c) $f_{position} = 0.5$, $f_{radius} = 1.0$, $f_{weak} = 1.0$, $f_{arrow} = 0.5$, (d) $f_{position} = 0.5$, $f_{radius} = 1.0$, $f_{arrow} = 0.1$.

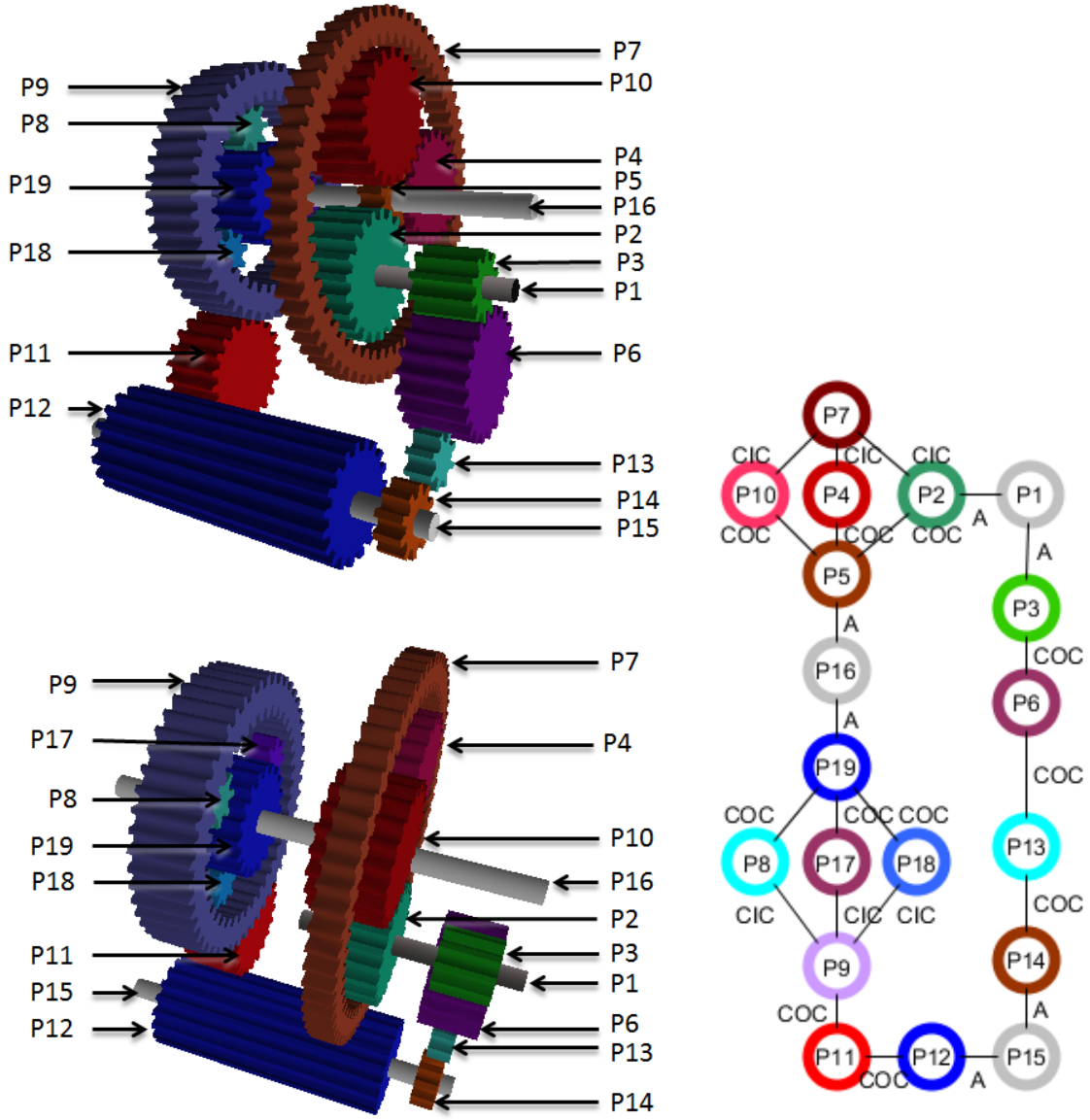


(a) (b)



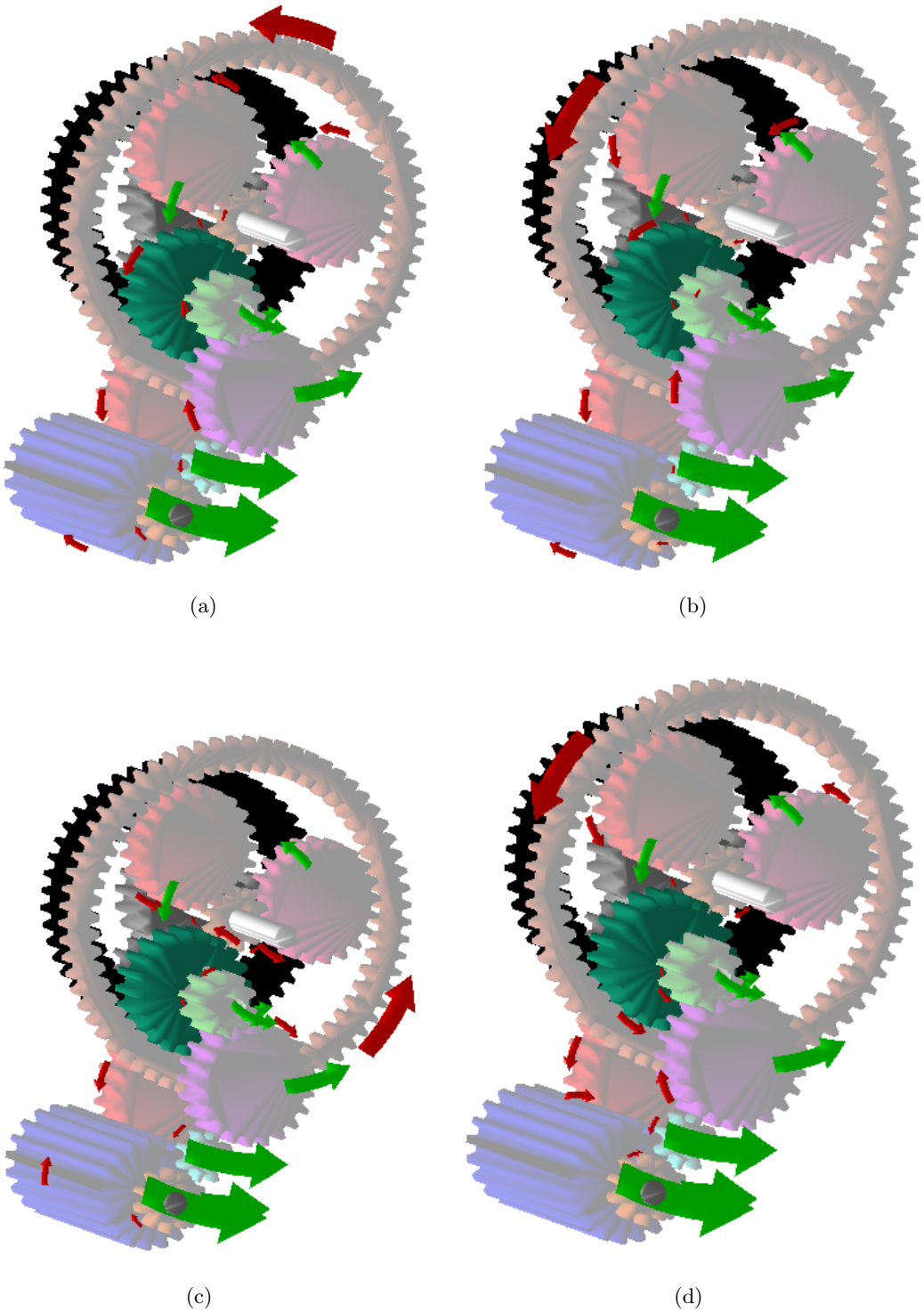
(c)

Obrázek C.4: (a) Testovací soustava *test1* a její graf interakce (b). Typy spojení jsou CIC (válec ve válci), COC (válec na válci) a A (osa). (c) Pro příklad vytvořené statické ilustrace se šipkami jsou uvedeny faktory, se kterými byla vytvořena: $f_{arrow} = 0.5$, $f_{position} = 0.5$, $f_{radius} = 1.0$, $f_{slice} = 1.0$.

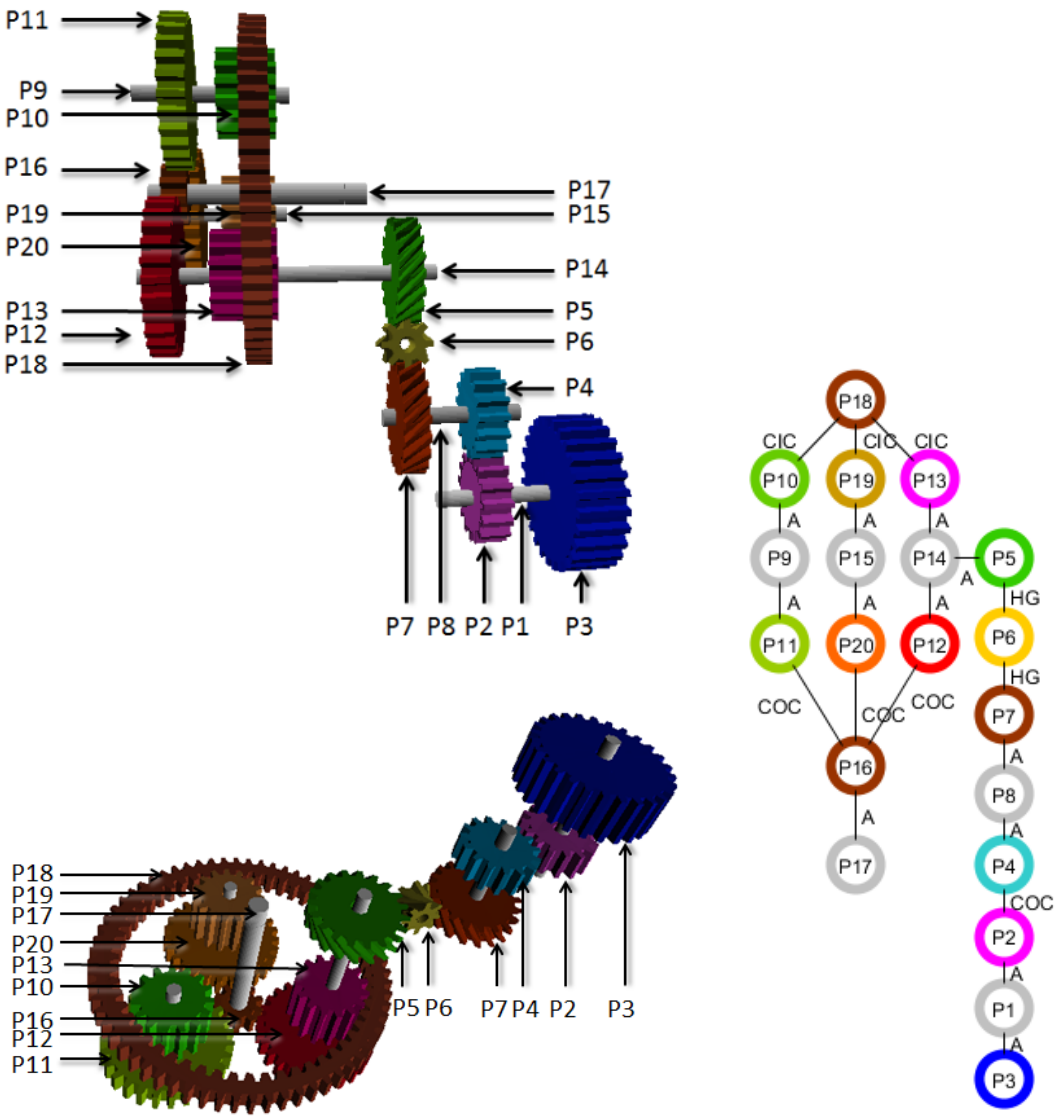


(a) (b)

Obrázek C.5: (a) Testovací soustava *test9* a její graf interakce (b). Typy spojení jsou CIC (válec ve válci), COC (válec na válci) a A (osa).

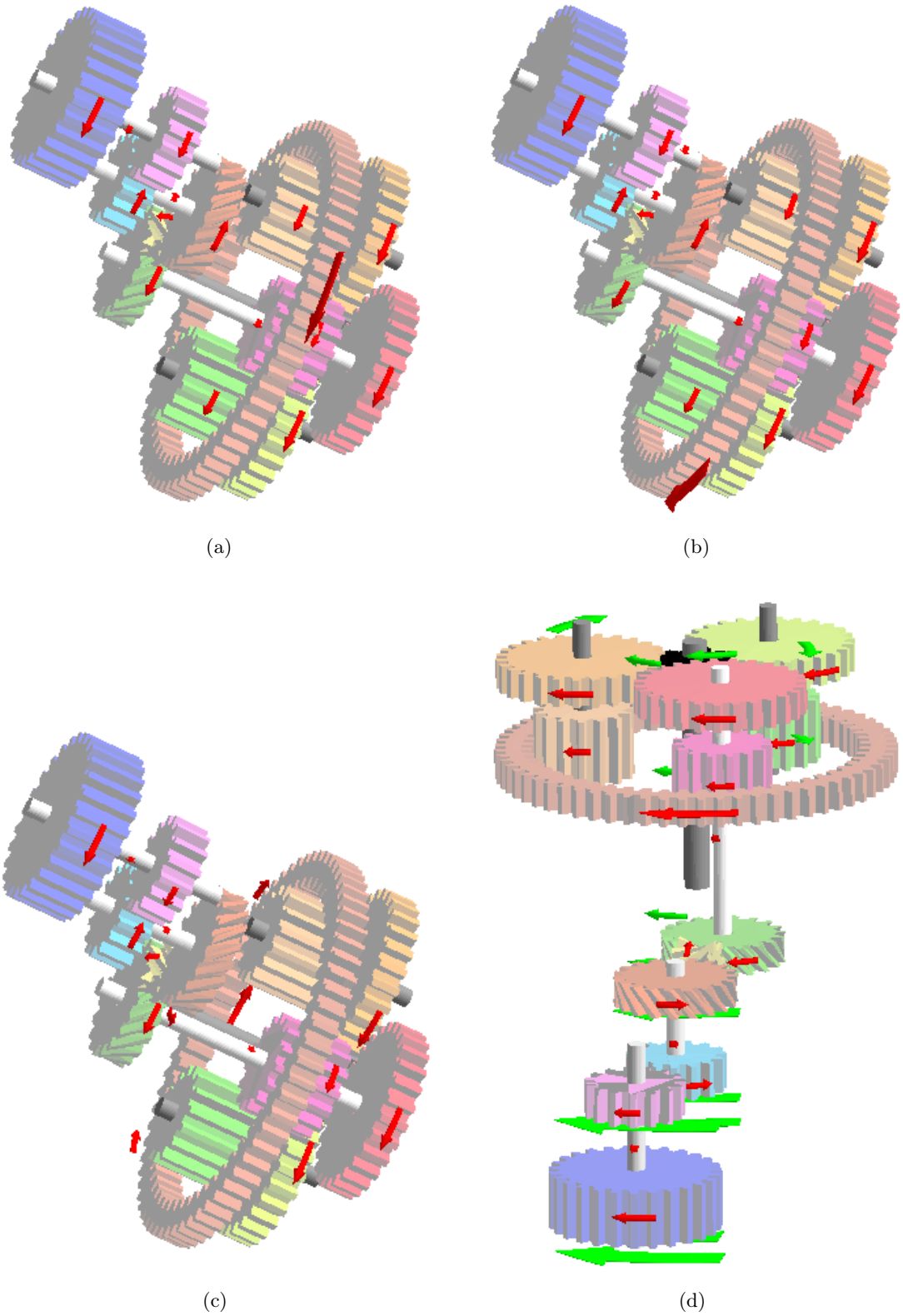


Obrázek C.6: Příklad statických ilustrací se šipkami testovací soustavy *test9*, které byly vytvořeny s následujícími hodnotami faktorů: $f_{position} = 0.5$, $f_{arrow} = 0.2$, $f_{radius} = 1.0$, $f_{slice} = 1.0$, (a) $f_{weak} = 1.0$; (b); (c) $f_{force} = 1.0$ (odpudivá síla); (d) $f_{force} = 1.0$ (přitažlivá síla).

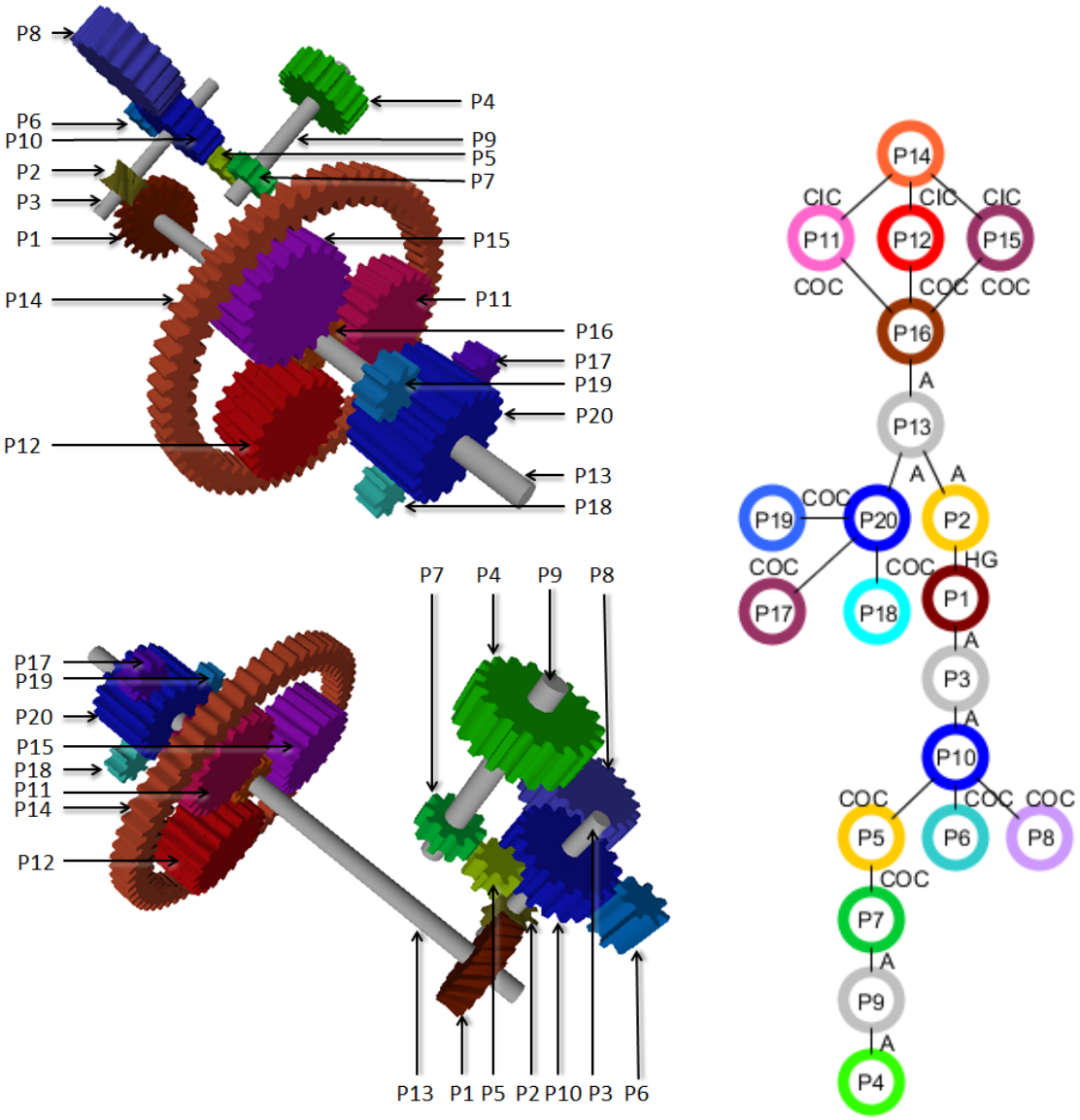


(a) (b)

Obrázek C.7: (a) Testovací soustava *test10* a její graf interakce (b). Typy spojení jsou HG (spirálové ozubené soukolí), CIC (válec ve válci), COC (válec na válci) a A (osa).

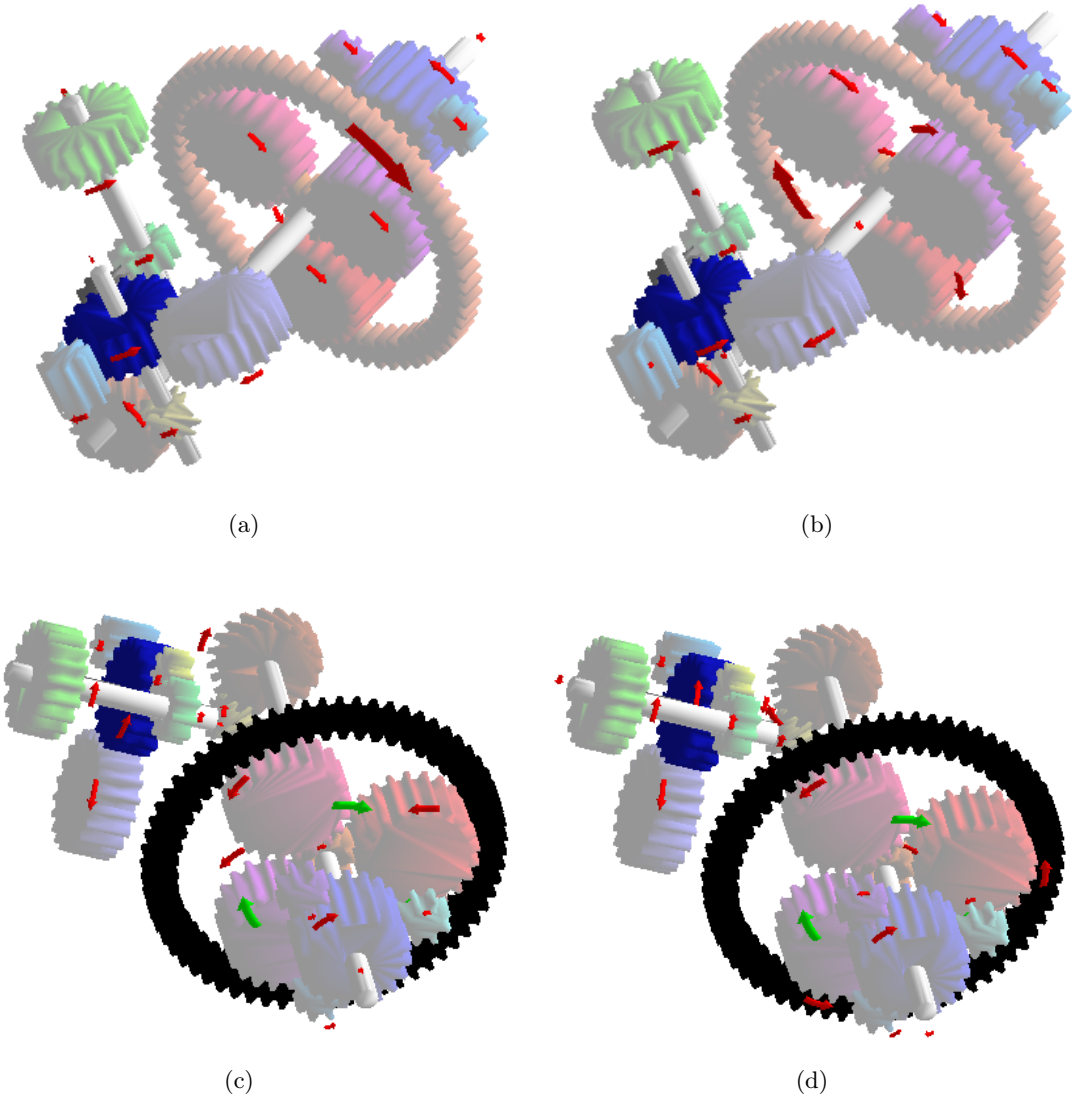


Obrázek C.8: Příklad statických ilustrací se šipkami testovací soustavy *test10*, které byly vytvořeny s následujícími hodnotami faktorů: $f_{position} = 0.5$, $f_{radius} = 1.0$, $f_{slice} = 1.0$, (a) $f_{arrow} = 0.5$, (b) $f_{weak} = 1.0$, (c) $f_{force} = 1.0$ (přitažlivé síly), $f_{arrow} = 0.1$, (d) $f_{arrow} = 0.5$.

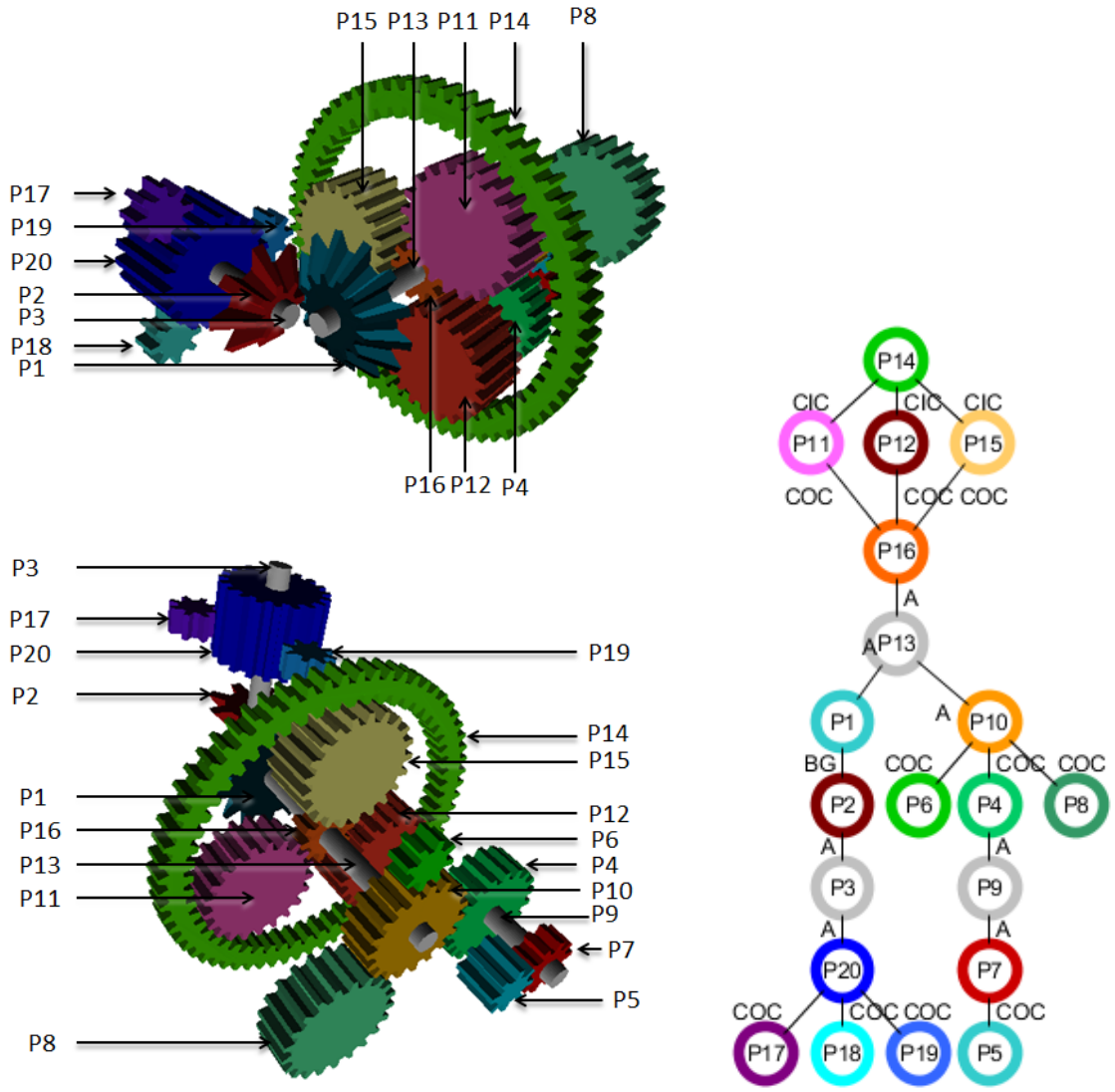


(a) (b)

Obrázek C.9: (a) Testovací soustava *test11* a její graf interakce (b). Typy spojení jsou HG (spirálové ozubené soukolí), CIC (válec ve válci), COC (válec na válci) a A (osa).

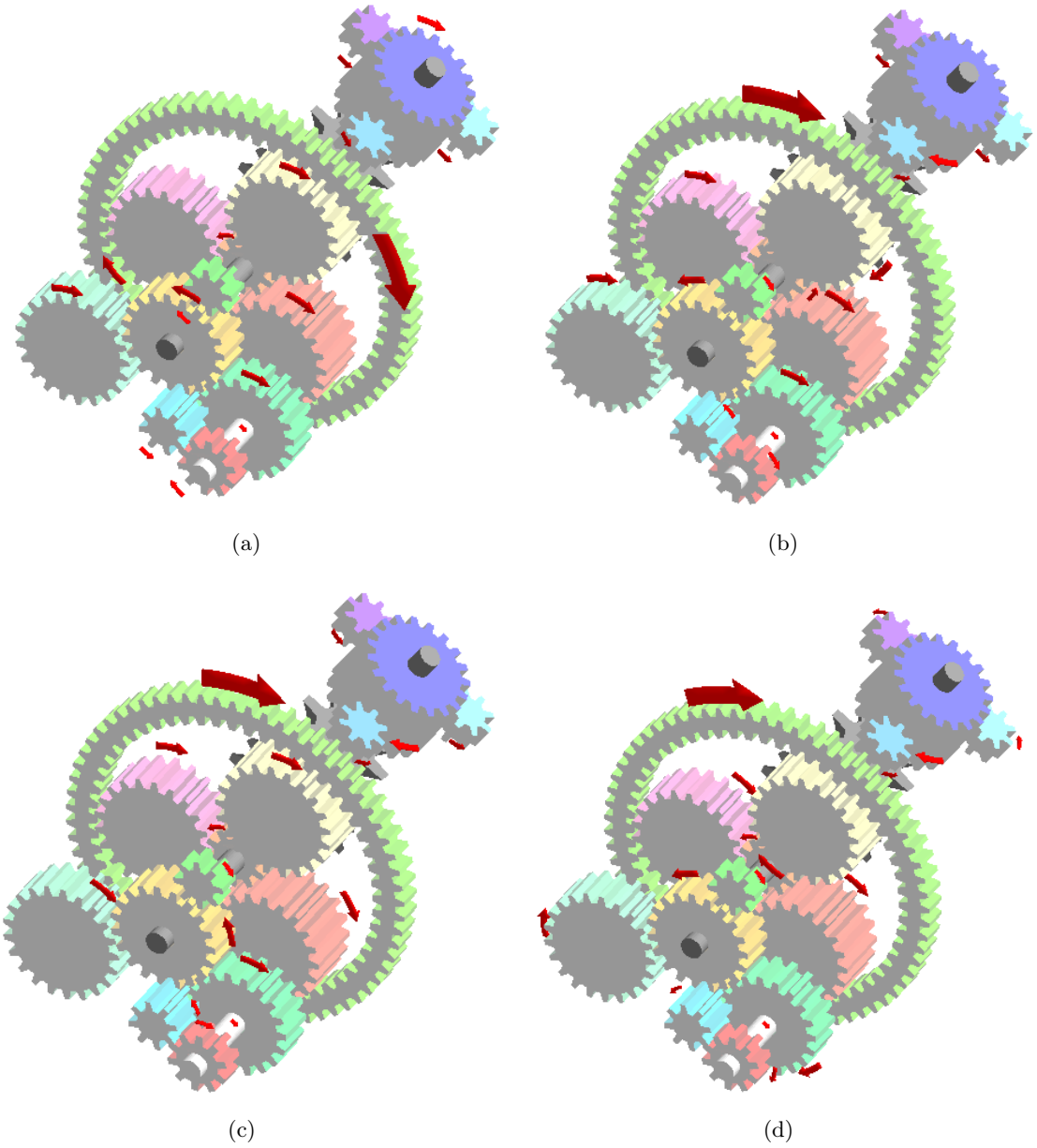


Obrázek C.10: Příklad statických ilustrací se šipkami testovací soustavy *test11*, které byly vytvořeny s následujícími hodnotami faktorů: $f_{position} = 0.5$, $f_{radius} = 1.0$, (a) $f_{arrow} = 0.5$, (b) $f_{slice} = 1.0$, $f_{arrow} = 0.1$, (c) $f_{slice} = 1.0$, $f_{arrow} = 0.3$, (d) $f_{slice} = 1.0$, $f_{force} = 1.0$ (přitažlivá síla), $f_{arrow} = 0.3$.

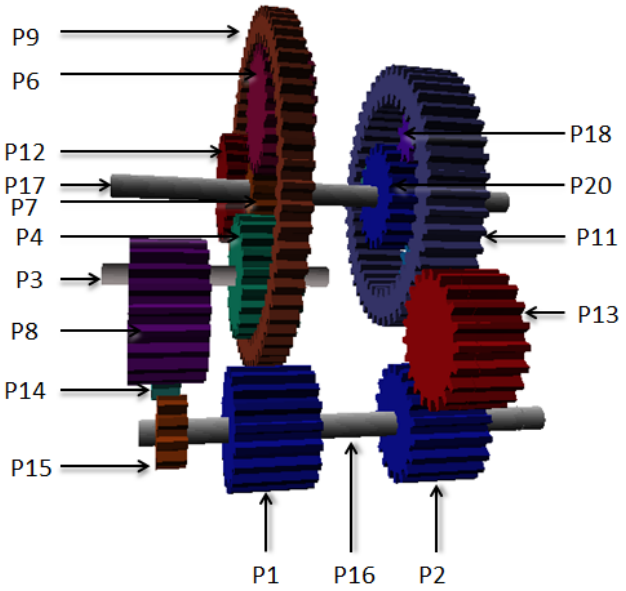


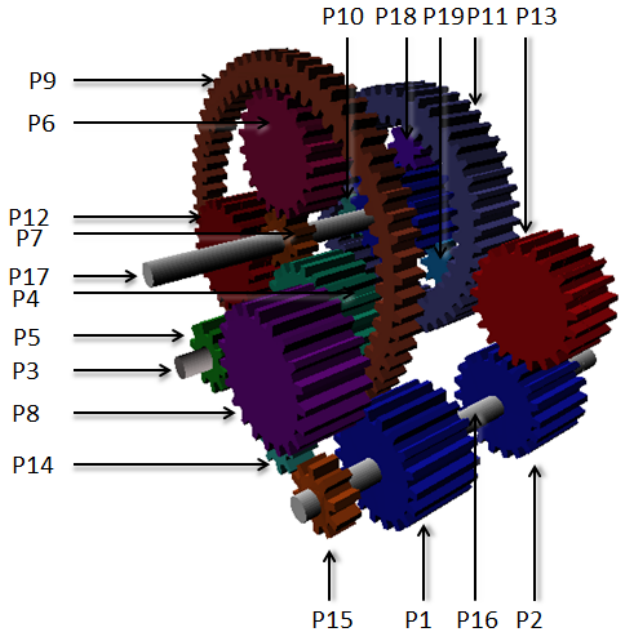
(a) (b)

Obrázek C.11: (a) Testovací soustava *test12* a její graf interakce (b). Typy spojení jsou BG (kuželové ozubené soukolí), CIC (válec ve válci), COC (válec na válci) a A (osa).

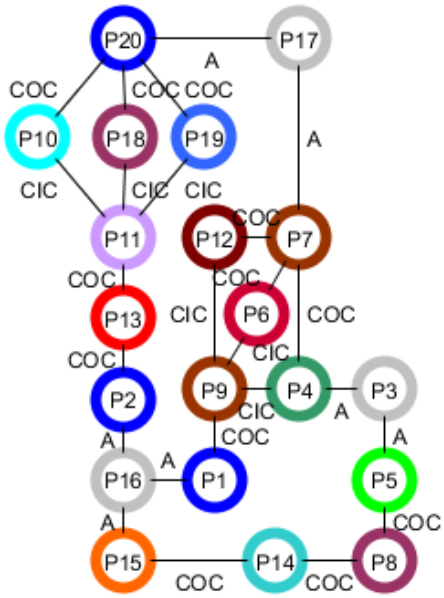


Obrázek C.12: Příklad statických ilustrací se šipkami testovací soustavy *test12*, které byly vytvořeny s následujícími hodnotami faktorů: $f_{position} = 0.5$, $f_{radius} = 1.0$, (a) $f_{arrow} = 0.5$, (b) $f_{slice} = 1.0$, $f_{arrow} = 0.5$, (c) $f_{slice} = 1.0$, $f_{force} = 1.0$ (přitažlivé síly), $f_{arrow} = 0.5$, (d) $f_{slice} = 1.0$, $f_{force} = 1.0$ (odpudivá síla), $f_{arrow} = 0.1$.



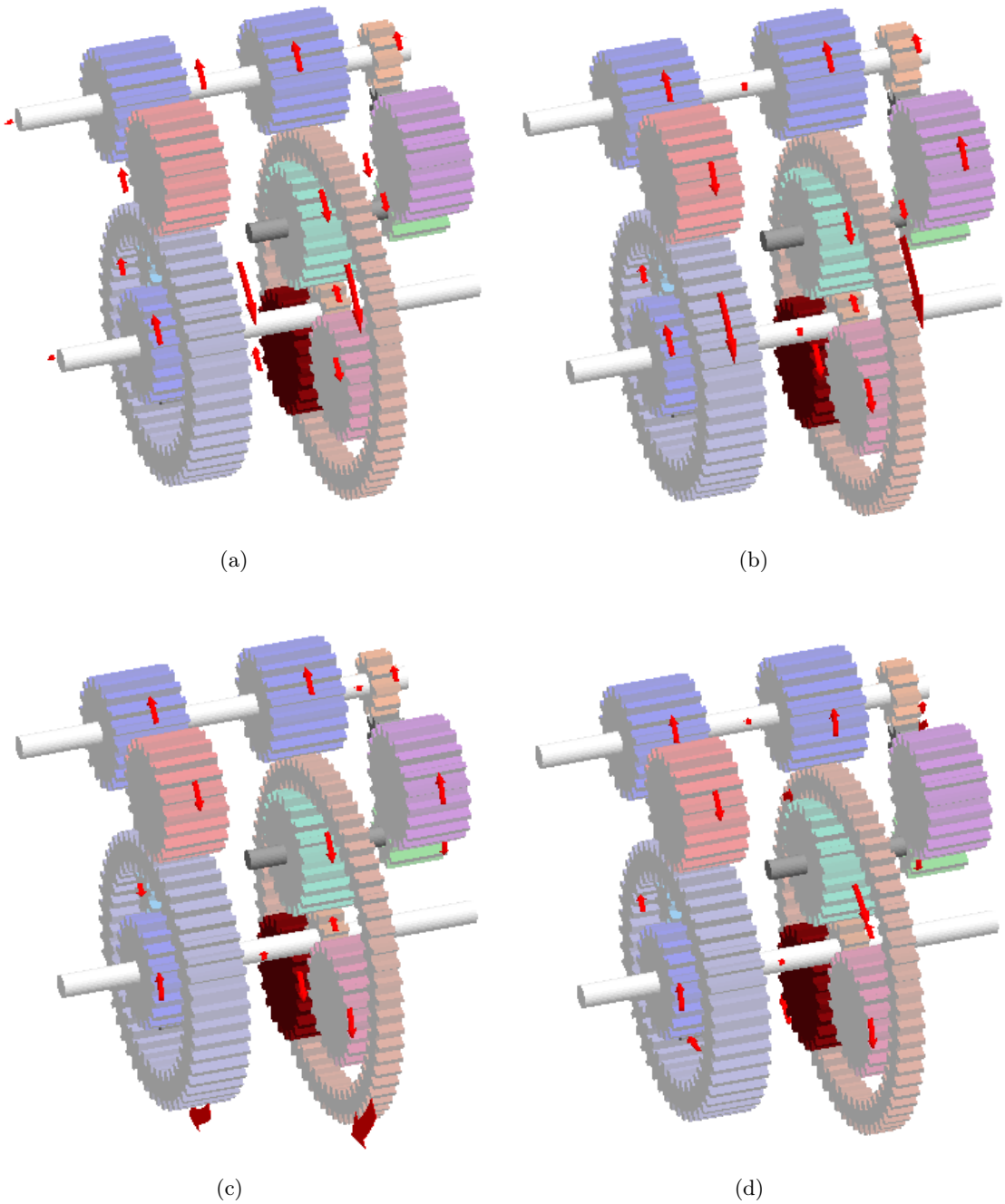


(a)



(b)

Obrázek C.13: (a) Testovací soustava *test13* a její graf interakce (b). Typy spojení jsou CIC (válec ve válci), COC (válec na válci) a A (osa).



Obrázek C.14: Příklad statických ilustrací se šipkami testovací soustavy *test13*, které byly vytvořeny s následujícími hodnotami faktorů: $f_{position} = 0.5$, $f_{radius} = 1.0$, (a) $f_{arrow} = 0.5$, (b) $f_{slice} = 1.0$, $f_{arrow} = 0.5$, (c) $f_{slice} = 1.0$, $f_{weak} = 1.0$, $f_{arrow} = 0.2$, (d) $f_{slice} = 1.0$, $f_{force} = 1.0$ (přitažlivá síla), $f_{arrow} = 0.1$.

Příloha D

Obsah přiloženého CD

Přiložené CD obsahuje dva adresáře: *realizace* a *text*. Zdrojové kódy, data, výsledky a použité knihovny jsou v adresáři *realizace*. Text práce ve formátu PDF a její TeX zdrojové soubory jsou uloženy v adresáři *text*.

Adresář realizace obsahuje pět podadresářů: doc, results, src, libraries, data, exe.

doc V tomto adresáři je obsažena vygenerovaná html verze dokumentace zdrojových kódů v programu Doxygen.

results Tento adresář obsahuje vytvořené dynamické vizualizace pohybu mechanických součástí soustavy, které byly zaznamenány do videa. Videa byla vytvořena s použitím programu CamStudio[8]. Pro spuštění vytvořených videí je potřeba mít nainstalován program CamStudio a jeho kodek.

src V tomto adresáři je uložen projekt, který byl vytvořen v Microsoft Visual Studiu 2010 se zdrojovými kódy této práce. Významné třídy a jejich metody byly popsány v podkapitole 3.

libraries V tomto adresáři jsou uloženy použité knihovny: assimp, glew, glm, jama, tnt, AntTweakBar, glut, Boost, CGAL. Popis těchto knihoven, jejich použité verze a jejich požadavky na hardware jsou uvedeny v příloze A.

data V tomto adresáři jsou v jednotlivých podadresářích obsaženy testovací soustavy mechanických součástí.

exe Tento adresář obsahuje přeloženou spustitelnou verzi programu a přiložené *dll* použitých knihoven.