

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
MULTIMEDIJOS INŽINERIJOS KATEDRA

Kęstutis Jankauskas
Tautvydas Perminas

**Gaminio modelio analizė ir koregavimas pritaikant
CAM sistemoms**
Magistro darbas

Kalbos konsultantė

Lietuvių k. katedros lekt.
Ilona Mickienė
2006-05-22

Vadovas

Dr. Giedrius Liutkus
2006-05-29

Recenzentas

Dr. doc. Jonas Matickas
2006-05-29

Atliko

IFM-0/1 gr. stud.
Kęstutis Jankauskas
Tautvydas Perminas
2006-05-29

Kaunas, 2006

TURINYS

1.	ĮVADAS	4
2.	GAMINIO MODELIO APDOROJIMO ANALIZĖ	7
2.1.	GAMINIO MODELIO APDOROJIMO SRITIES APŽVALGA	7
2.2.	GAMINIO MODELĮ APDOROJANČIOS PROGRAMINĖS ĮRANGOS FUNKCIJOS	11
2.3.	GAMINIO MODELIO APDOROJIMO LITERATŪROS ANALIZĖ	12
2.4.	CAD/CAM RINKOS ANALIZĖ	14
2.4.1.	<i>Programinės įrangos tobulėjimo tendencijos</i>	14
2.4.2.	<i>Duomenų bylų formatai</i>	15
2.4.3.	<i>Apibendrinimas</i>	16
2.5.	NURBS (NEUNIFIKUOTO RACIONALIOJO B-SPLAINO) APRAŠYMAS	18
2.6.	IGES BYLŲ FORMATO APRAŠYMAS	23
2.6.1.	<i>IGES struktūra</i>	23
2.6.2.	<i>Pradžios sekcija S</i>	24
2.6.3.	<i>Globalioji sekcija G</i>	25
2.6.4.	<i>Adresų įrašų sekcija D</i>	26
2.6.5.	<i>Parametrų sekcija P</i>	28
2.6.6.	<i>Pabaigos sekcija T</i>	28
2.6.7.	<i>Apibendrinimas</i>	29
2.7.	OPENGL NURBS APRAŠYMAS	31
2.7.1.	<i>NURBS kreivė</i>	31
2.7.2.	<i>NURBS paviršius</i>	32
2.7.3.	<i>Iškarpytasis NURBS paviršius</i>	33
2.8.	TŪRIO SKAIČIAVIMO ALGORITMAS	35
2.9.	MODELIO KOREGAVIMO (NURODYTOJE SRITYJE) ALGORITMAS	39
2.10.	PAKAVIMO DVIMATĖJE ERDVĖJE ALGORITMAS	42
2.11.	DVIMAČIO KONTŪRO KOREGAVIMO ALGORITMAI	46
2.11.1.	<i>Kontrolinių taškų kiekio minimizavimas</i>	46
2.11.2.	<i>Splaininės kreivės vertimas į atkarpas ir lankus</i>	48
3.	GAMINIO MODELIO APDOROJIMO PROJEKTAVIMAS	50
3.1.	REIKALAVIMŲ SISTEMAI SPECIFIKACIJA	50
3.2.	DUOMENŲ STRUKTŪRA	51
3.2.1.	<i>Gaminio modelio duomenys</i>	51
3.2.2.	<i>Pagalbiniai duomenys</i>	54
3.3.	SISTEMOS ARCHITEKTŪRA	56
3.4.	PROGRAMINIŲ MODULIŲ IR OBJEKTŲ SPECIFIKACIJOS	58
3.4.1.	<i>„Parsing“ modulis</i>	58
3.4.2.	<i>„GetEntities“ modulis</i>	60

3.4.3.	„DrawSurface“ modulis.....	61
3.4.4.	„Packing“ modulis.....	61
3.4.5.	„ModelEditing“ modulis	62
3.5.	TESTAVIMO MEDŽIAGA	63
3.5.1.	Duomenų iš IGES bylos skaitymas ir rašymas	63
3.5.2.	IGES bylos turinio interpretavimas ir vaizdavimas.....	63
3.5.3.	Tūrio skaičiavimas.....	65
3.5.4.	Pasirinktos modelio srities proporcijų koregavimas	66
3.5.5.	Pakavimas dvimatėje erdvėje	67
3.5.6.	NURBS kreivės koregavimas	68
4.	VARTOTOJO DOKUMENTACIJA	71
4.1.	FUNKCINIS SISTEMOS „ASSISTANT3D“ APRAŠYMAS	71
4.2.	SISTEMOS „ASSISTANT3D“ VADOVAS	72
4.2.1.	Vartotojo sąsaja.....	72
4.2.2.	Funkcijų sąrašas.....	73
4.2.3.	Funkcijų aprašymas.....	76
4.2.4.	Klaidų pranešimai	79
4.3.	SISTEMOS „ASSISTANT3D“ ĮDIEGIMO VADOVAS	82
5.	IŠVADOS.....	83
6.	LITERATŪRA.....	84
7.	TERMINŲ IR SANTRUMPŲ ŽODYNAS.....	86
8.	PRIEDAI	88
8.1.	IGES FORMATO ESYBIŲ TIPŲ APRAŠYMAS	88
8.1.1.	Apskritas lankas (100).....	89
8.1.2.	Atkarpa (110).....	90
8.1.3.	Transformacijų matrica (124)	91
8.1.4.	Racionalioji B-splaininė kreivė (126).....	92
8.1.5.	Racionalusis B-splaininis paviršius (128)	94
8.1.6.	Ribojančioji esybė (141).....	97
8.1.7.	Kreivė parametritiniame paviršiuje (142)	101
8.1.8.	Apribotasis paviršius (143).....	103
8.1.9.	Iškarpytasis (parametrinis) paviršius (144)	104
8.1.10.	Spalvos esybė (314)	106

1. ĮVADAS

Šio darbo tikslas yra ištirti *CAD* (*Computer Aided Design*) / *CAM* (*Computer Aided Manufacturing*) programinių paketų rinką ir sukurti programinę įrangą (PI), užpildančią specifines funkcionalumo spragas bei lengvinančią projektuotojų darbą.

Paprastai *CAD* ir *CAM* sistemos naudoja tuos pačius metodus objektų geometrinei konstrukcijai aprašyti. Tačiau *CAD* programinės įrangos funkcijos skirtos patogiam geometrinių objekto kūrimui ir redagavimui. Šie įrankiai pranoksta *CAM* programinių paketų įrankius. Pastarieji leidžia atlikti primityvius geometrijos koregavimo veiksmus, bet turi visas būtinas funkcijas apdirbimo programų rašymui, kurios vėliau valdo gamybos stakles. Gamybos rezultatas yra gaminys, o jo geometrinę konstrukciją atspindinti kopija virtualioje erdvėje (*CAD/CAM* paketuose) vadinama gamino modeliu.

Šio darbo tikslas yra sukurti PI „*Assistant3D*“, galinčią analizuoti ir redaguoti gaminio modelį, paruošiant jį apdorojimui *CAM* sistema. Toliau gaminio modelio analizė ir redagavimas bus vadinami bendresniu, bet trumpesniu terminu – „gaminio modelio apdorojimas“. PI funkcijos turėtų kompensuoti *CAD* ir *CAM* tarpusavio trūkumus, įranga turėtų būti specifinis minėtų sistemų papildymas, kuris veiktų savarankiškai (ne kaip kitos PI priedas). Programos duomenų šaltiniais galėtų būti tiek *CAD*, tiek *CAM* programiniai paketai, tačiau programos darbo rezultatas turi būti paruoštas apdorojimui *CAM* sistema (konkrečiai *MasterCAM v 6.0* ar vėlesne). Programos apimtis nėra apribota ir gali būti plečiama.

Atsižvelgiant į PI užsakovų pageidavimus ir pastebėjimus, daromos prielaidos, jog standartinis *CAM* sistemos komplektas neturi galimybės greitai ir patogiai skaičiuoti splaininių paviršių ribojamos erdvės tūrio. Priedai prie sistemos, kurie galėtų būti naudojami tūrio skaičiavimui, nėra skirti šiai specifinei užduočiai ir turi daug kitų funkcijų, todėl labai brangūs ir griezdiški. Taip pat *CAM* sistema neturi patogių įrankių, skirtų gaminio modelio modifikavimui trimatėje erdvėje. Šią problemą būtų galima spręsti įsigyjant naują *CAD* sistemą, tačiau tokia PI brangiai kainuotų, o personalo mokymui būtų išseikvota nemažai laiko. Be to, dėl tikslinės užsakovo darbo srities, labai svarbi galimybė redaguoti modelį ir tuo pačiu stebėti jo tūrio kitimą.

Šių uždavinių sprendimas aktualus ne tik konkrečiam užsakovui, bet ir kitiems panašia veikla besiverčiantiems gamintojams, kurie gamybos procesuose naudoja kompiuterizuotas sistemas. Dėl šios priežasties svarbu, jog PI naudojami duomenų formatai skirti tiek informacijos skaitymui, tiek rašymui būtų kuo universalesni, funkcionalumas neapribotas ir lengvai plečiamas, patogi ir lengvai suprantama vartotojo sąsaja.

Ankščiau paminėtos trūkstamos funkcijos trimatėje erdvėje (3D). Taip pat į „Assistant3D“ funkcijų sąrašus įtrauktas dvimačių (2D) kontūrų apdorojimas. 3D ir 2D funkcijos operuoja skirtingo tipo objektais ir negali manipuliuoti kito tipo objektais. 3D ir 2D funkcijų aibės nepersidengia, todėl šios dvi grupės griežtai atskirtos.

Atsižvelgiant į visus aptartus faktorius, buvo pasirinktos konkrečios PĮ paketo funkcijos:

Trimatėje erdvėje:

1. *IGES* formato nuskaitymas.
2. Trimatis modelio vaizdas.
3. Galimybė keisti modelio proporcijas atskirai kiekvienai koordinačių sistemos ašiai, nurodytoje modelio srityje.
4. Uždaros paviršiais apribotos erdvės tūrio skaičiavimas.
5. Duomenų saugojimas *IGES* formatu.

Dvimatėje erdvėje:

1. *IGES* formato nuskaitymas.
2. Kontūrų vaizdavimas.
3. Netaisyklingų neiškilių uždarų kontūrų pakavimas duotame plote, paruošiant gaminio modelį apdorojimui *CAM* paketu.
4. Kontūro koregavimas:
 - a. Kontrolinių taškų skaičiaus minimizavimas.
 - b. Splaino vertimas į geometrines primityvas (atkarpas ir lankus).
5. Duomenų saugojimas *IGES* formatu.

Visų funkcijų realizavimo metodai, duomenų formatai aprašyti 2 skyriuje.

Naudotos literatūros sąrašą galima rasti 6 dalyje. 2 skyriuje išsamiai nagrinėjami moksliniai veikalai, knygos, straipsniai ir ištraukos konferencijų susijusių su darbo tema. Aptariamose analizėse metu nustatytos PĮ *CAD/CAM* rinkoje tobulėjimo tendencijos, trūkumai, populiarūs ir naudotini duomenų formatai, siūlomi įrankiai ir algoritmai, aktualiems uždaviniams realizuoti. Taip pat pateikiami nauji, ankščiau nepublikuoti algoritmai.

Gaminio modelis turi būti labai tikslus (gaminančios mašinos tikslumas gali siekti 0,01 mm), todėl įprastinės vaizdiniam projektavimui skirtos modelio konstrukcijos ir duomenų formatai nėra tinkami. Dėl šios priežasties paviršiams ir kreivėms išreikšti buvo pasirinktas *NURBS* (*Non-Uniform Rational B-Spline*) splainas (žr.: 2.5 skyrius). Duomenų saugojimas, naudojant *NURBS*, labai kompaktiškas, o modelio geometrija gali būti apskaičiuojama bet koku tikslumu.

Duomenų nuskaitymui ir išsaugojimui pasirinktas *IGES* (*Initial Graphics Exchange Specification*) (žr.: 2.6 skyrius). Nors jis atsirado praeito amžiaus pačioje devinto dešimtmečio

pradžioje, tačiau sėkmingai naudojamas daugelio *CAD/CAM* sistemų iki šių dienų. Nors šis formatas buvo skirtas duomenų pernešimui tarp visų skirtingų to meto *CAD/CAM* sistemų ir apima labai plačią sferą, labiausiai dominanti jo savybė yra patogiai saugoti *NURBS* aprašus (*ASCII* formatu). Šios bylos yra lengvai suprantamos ir redaguojamos *CAM* srities ekspertų.

Gaminio modelis vaizduojamas, naudojant *OpenGL* (žr.: 2.7 skyrius) trimatės grafikos vaizdavimo realiuoju laiku variklį, dėl galimybės vaizduoti *NURBS* paviršius ir kreives, naudojant duomenis, nuskaitytus iš *IGES* bylos. Duomenų formatai puikiai suderinti, todėl nuskaičius, užtenka minimaliai apdoroti, kad būtų paruošti vaizdavimui.

Darbe naudojamos programos:

- *MasterCAM v 6.0* ir vėlesnės versijos – pagrindinė *CAM* PĮ, atstovaujanti duomenų šaltinį ir įrangą, skirtą gautam rezultatui apdoroti. Jos pagalba bus testuojamas kuriamos PĮ sugeneruotų bylų teisingumas.
- *3D Studio Max R 5.0* ir vėlesnės versijos – PĮ atstovaujanti *CAD* sistemą, nors ji nėra skirta tiksliai gaminio modeliui sudaryti ir daugiau skirta vaizdiniam projektavimui, tačiau turi daug įrankių sudėtingam modelio modifikavimui. Šios sistemos generuojami modeliai ir juos aprašančios duomenų struktūros gali būti daug sudėtingesni bei išnaudojantys daugiau galimybių nei *MasterCAM v 6.0*.
- *Borland C++ Builder 6* – *C++* programavimo kalbos redaktorius ir kompiliatorius. naudojamas kuriamos programinės įrangos realizacijai.
- *OpenGL* – trimatės grafikos vaizdavimo realiuoju laiku variklis, naudojamas gaminio modelio vaizdavimui.

2. GAMINIO MODELIO APDOROJIMO ANALIZĖ

2.1. *Gaminio modelio apdorojimo srities apžvalga*

Gaminio šiame darbe vadinamas produktas tokio kompiuterizuoto gamybos proceso, kurio eiga aprašyta, naudojant *CAM* sistemą. Gaminio modeliu vadinama virtuali gaminio kopija, kuria remiantis, *CAM* sistema sudaro gamybos proceso eigą. Gaminio modelis gali būti sukurtas tiek su *CAD*, tiek su *CAM* įrankiais. Modelį sudaro geometrinių objektų visuma. Geometrinio objekto konstrukcija privalo būti sąlyginai paprastai aprašoma, apskaičiuojama bet kokių tikslumu, taupanti kompiuterio atmintį. Todėl paviršių ir kreivių konstrukcijos aprašomos, naudojant *NURBS*. Išskirtinių atveju trimatis objektas gali būti geometrinis primityvas (sfera, cilindras). Tokie geometriniai objektai yra aprašomi keliais pozicijos ir matmenų parametrais (*Solid* geometrija), ir šiame darbe nenagrinėjami.

CAD (*Computer Aided Design*) – programinis paketas, skirtas tiksliems geometriniais modeliams sudaryti. Šių modelių paskirtis gali būti įvairi: vaizdinis pateikimas, gaminio modelis, architektūrinis projektas ir t.t. *CAD* programiniu paketu sukurtas modelis neretai būna naudojamas kitų sistemų, turinčių specifinį praktinį taikymą (gamyba, fizinių reiškinių simuliacija). *CAD* sistemos įrankiai, skirti modelio kūrimui ir redagavimui. Šie įrankiai patogesni ir turi daugiau galimybių, nei kitų sistemų su specializuota taikymo sritim. Tačiau vienintelė tokios sistemos praktinė realizacija yra modelį aprašančios konstrukcijos duomenų saugojimas ir vaizdinis modelio pateikimas.

Vaizdinis pateikimas paprastai nereikalauja didelio tikslumo, tačiau geometriniai objektai padengiami tekstūromis, jiems suteikiama specifinę medžiagą imituojantys parametrai. Papildomai skaičiuojamas šviesos šaltinių poveikis objektams, šešėliai, atspindžiai. Taip išgaunamas vadinamasis „foto-realistinis“ vaizdas. Modeliui, aprašančiam gaminį, „foto-realistiškumas“ visiškai nereikalingas, tačiau modelio tikslumas čia labai svarbus. Todėl sudėtingiems geometriniais objektams naudojama praktiškai neribotą tikslumą leidžia pasiekti splainų technologija.

Labiausiai paplitusios *CAD* sistemos: *AutoCAD*, *ArchiCAD*. *Maya* ir *3D Studio Max* irgi būtų galima priskirti prie *CAD* sistemų, nors jos specializuotos ne sudaryti tikslų modelį, o pateikti „foto-realistišką“ vaizdą ar animaciją.

CAM (*Computer Aided Manufacturing*) – programinis paketas, skirtas gaminio apdirbimo keliams sudaryti, kurie vėliau perrašomi į gamybos mašinas valdančias programas. Apdirbimo

keliai sudaromi, atsižvelgiant į gaminio modelio geometrinę konstrukciją, konkrečios gamybinės mašinos parametrus, laiko resursus, medžiagą, iš kurios gaminamas produktas. *CAM* sistemos taip pat turi keletą įrankių geometrinių objektų kūrimui ir modelio redagavimui, bet jie daug primityvesni už tokio tipo įrankius *CAD* sistemose.

Priklausomai nuo gaminio paskirties, jis gali būti aprašomas *2D*, *3D* koordinatėmis ar homogeninėmis *4D* koordinatėmis. Taip pat gamybos mašinos gali būti skirstomos pagal ašių skaičių. Vienos iš paprastesnių funkcionuoja trijų ašių pagrindu (ašys atitinka gaminio modelio erdvės Ox , Oy ir Oz koordinatinių ašis). Sudėtingesnės turi keturias ar penkias ašis (įrankis gali judėti Ox , Oy , Oz kryptimis, sukstis aplink savo pagrindą ir lankstytis apie ašį statmeną pagrindui). Tokios staklės turi daugiau galimybių, tačiau ir programos valdančios mašiną sudėtingesnės.

Gaminio modelis, kuriuo remiantis rašomos apdirbimo programos, sudaromas naudojant *Solid* geometriją arba splainus. Dėl savo populiarumo paprastai naudojami *NURBS* splainai.

Gerai žinomos *CAM* sistemos: *MasterCAM*, *EZCAM*, *SolidCAM*, *PowerMILL*.

NURBS (Non-Uniform Rational B-Spline) – neunifikuoto racionaliojo B-splaino pagrindas yra *Bezier* kreivė. B-splainas sudarytas iš kelių *Bezier* segmentų su tam tikru tęstinumo laipsniu (žr.: 2.5 skyrius), lokalizuojant kontrolinių taškų poveikį kreivei bei pridėdant svorius. Tai leidžia realizuoti ir antros eilės kreives (apskritimus, elipses, paraboles ir hiperboles).

Splaininė konstrukcija tikslumo ir kompaktiškumo santykio atžvilgiu turi didelį pranašumą prieš poligoninę konstrukciją. Dėl galimybės skaičiuoti splaino reikšmes bet koku žingsniu neprarandant tikslumo, šios konstrukcijos plačiai naudojamos *CAM* sistemose. Poligoninė konstrukcija tinkama tik vaizdiniam pateikimui, bet ne gaminio modelio aprašymui, nes didelio tikslumo modeliui sudaryti reikėtų labai daug taškų.

Šiame darbe gaminio modelio apdorojimas skirstomas į modelio analizę ir koregavimą. Analizės funkcijos: paviršiais apribotos erdvės tūrio apskaičiavimas. Koregavimo funkcijos: modelio proporcijų keitimas nurodytoje modelio srityje, kontūrų pakavimas duotoje erdvėje (minimizuojant ploto sąnaudas), kolinearinių taškų skaičiaus minimizavimas, splainų vertimas į lankus bei atkarpas (žr.: 2.2 skyrius).

Tūrio skaičiavimas. Dauguma *CAD/CAM* sistemų gali suskaičiuoti primityvių geometrinių objektų (*Solid* geometrijai) tūrius, įstatydamos juos aprašančių parametrų reikšmes į tūrio skaičiavimo formules. Tačiau jei gaminio modelį sudaro ne primityvūs objektai, o netaisyklingi ir sudėtingi paviršiai, tada toks skaičiavimo būdas netinka. Vienas iš metodų, kuriuo galima apskaičiuoti erdvės, apribotos paviršiais, tūrį, yra apdirbimo kelių modeliui sudarymas. Vėliau, pagal įrankio parametrus ir apdirbimo kelio ilgį bei konfiguraciją, galima apskaičiuoti erdvės, kurioje judėjo darbinė įrankio dalis, tūrį. Toks metodas reikalauja papildomų žinių apie apdirbimo

kelių sudarymą. Taip pat šiam procesui reikia daug laiko, kad suvesti įrankio ir gamybos mašinos parametrus, duoti atitinkamas komandas *CAM* sistemai, atsižvelgiant į paviršių konstrukciją, kad būtų inicijuotas apdirbimo kelių sudarymas. Stokojant patirties kiekvienas iš šių žingsnių gali būti klaidingo rezultato priežastimi. Be to, standartinis *CAM* komplektas gali ir neturėti reikiamų funkcijų. Papildomų modulių pirkimas padidintų išlaidas. Šio uždavinio sprendimas analizuojamas 2.8 skyriuje.

Proporcijų keitimas nurodytoje modelio srityje. Objekto, jo dalies ar objektų grupės matmenų keitimas vienos koordinačių ašies kryptimi, leidžia keisti proporcijas. Šis procesas realizuojamas, taikant specifinę transformacijų matricą pasirinktos modelio srities kontroliniams taškams atsparos taško atžvilgiu. Atsparos taškas gali būti pasirenkamas, paprastai jis būna geometriniam pasirinktos srities centre (taškų pozicijų vidurkis). Ši funkcija labai svarbi kartu su tūrio skaičiavimu. Kai kuriems gamintojams (butelių formų gamyba plastmasinės taros pūtimo mašinoms), gali būti labai svarbu pakoreguoti modelio ar jo srities proporcijas ir matyti tūrio pokyčius. Taip modelio forma gali būti koreguojama tol, kol pasiekiamas užduotas tūrio cenzas. Proporcijų koregavimo algoritmas aprašomas 2.9 skyriuje.

Kontūrų pakavimas. Pakavimas realizuojamas išdėliojant dvimates figūras lape su duotais matmenimis. Figūros gali būti netaisyklingos, neiškilos su skylėmis savo viduje (kontūras kontūre), tačiau negali savęs kirsti, išskyrus pabaigos ir pradžios taškus. Į figūrą aprašantį kontūrą gali įeiti atkarpos, lankai, splainai. Svarbu, kad jų pradžios ir pabaigos taškai jungtųsi ir sudarytų vientisą kontūrą. Be to, užduotas minimalus atstumas tarp figūrų. Šios funkcijos esmė paruošti kontūrų aibę apdorojimui *CAM* paketu taip, kad nereikėtų gaišti laiko, pozicionuojant figūras po vieną rankiniu būdu. Tarpai tarp figūrų reikalingi medžiagą apdirbančiam įrankiui su apibrėžtu diametru. Jei tarpas tarp figūrų mažesnis už diametrą, pirmosios figūros apdirbimo metu sugadinama antroji. Uždavinio sprendimas aprašomas 2.10 skyriuje.

Kreivės taškų skaičiaus minimizavimas. Jei kontūrą ar jo dalį aprašo kreivė, tada tikėtina, kad ji turės taškų perteklių – neefektyviai išnaudojama atmintis. Taip paprastai atsitinka duomenis perkeliant iš vienos sistemos į kitą, keitimo programų pagalba. Jei trys gretimi taškai sudaro dvi atkarpas, besikertančias viduriniajame taške, ir jų susikirtimo kampas artimas 180 laipsnių (skaičiuojama pagal užduotą paklaidos reikšmę), tada vidurinis taškas panaikinamas. Svarbu pastebėti, kad *NURBS* kreivėse dažnai naudojami trys taškai su tomis pačiomis koordinatėmis aštriam kampui sudaryti (jei splaino bazinės funkcijos laipsnis lygus 3). Į tai būtina atsižvelgti. Algoritmo analizė pateikta 2.11 skyriuje.

Splainų vertimas į lankus ir atkarpas. Prieš šią operaciją splainą aprašančių kontrolinių taškų skaičius turėtų būti minimizuotas. Atsižvelgiant į kontrolinių taškų pozicijas, splaino segmentai gali

būti pakeisti atkarpomis ar lankais. Kontūras, saugojamas kaip primityvių planimetrinių objektų seka užima daugiau vietos atmintyje, bet kai kuriems algoritmams toks pavidalas yra būtinas, nes pastarieji neparuošti darbui su splineais. Kontūrų pakavimo algoritmas automatiškai splaino kopiją paverčia linijom, kad galėtų dirbti toliau. Algoritmo analizė pateikta 2.11 skyriuje.

Literatūros, kuria buvo remiamasi darbe, apžvalga pateikta 2.3 skyriuje. Visi nedokumentuoti algoritmai ar jų dalys buvo sudaryti pačių autorių, remiantis sukauptomis žiniomis, patirtimi ir eksperimentais.

2.2. Gaminio modelį apdorojančios programinės įrangos funkcijos

Šiame skyriuje pateikiama paviršutiniška kuriamos programinės įrangos funkcijų analizė ir nuorodos į skyrius su detaliais aprašymais.

<u>Nr.</u>	<u>Funkcija</u>	<u>Trumpas aprašymas</u>	<u>Skyrius</u>
1.	Duomenų skaitymas ir rašymas į <i>IGES</i> formato bylą.	Nuskaitomi duomenys, aprašantys gaminio modelį iš <i>CAD/CAM</i> sistemos sugeneruotos <i>IGES</i> bylos ir saugomi PĮ duomenų bazėje. Apdoroti PĮ duomenys išsaugomi <i>IGES</i> byloje. Pagrindiniai geometrinių esybių tipai: atkarpa, lankas, <i>NURBS</i> (žr.: 2.5 skyrius) kreivė ir paviršius.	2.6
2.	Gaminio modelio vaizdavimas.	Duomenų bazėje esantys duomenys naudojami gaminio modelio geometriniai konstrukcijai vaizduoti su <i>OpenGL</i> . <i>NURBS</i> paviršiams ir kreivėms naudojama <i>GLU</i> biblioteka.	2.7
3.	Paviršiais apribotos erdvės tūrio apskaičiavimas.	Paviršiai ir pagalbinės plokštumos naudojamos uždarai trimatės erdvės sričiai nustatyti, kurios tūrį reikia apskaičiuoti. Tūris skaičiuojamas, atsižvelgiant į diskretizavimo žingsnio reikšmę.	2.8
4.	Modelio proporcijų keitimas nurodytoje modelio srityje.	Keičiamos paviršiaus(-ių) srities proporcijos pasirinktos koordinačių ašies kryptimi, manipuliuojant kontroliniais taškais. Atsparos taškas taip pat gali būti pasirenkamas.	2.9
5.	Kontūrų pakavimas duotoje erdvėje, minimizuojant sunaudojamą plotą.	Duota aibė kontūrų ir stačiakampė plokštuma, kurią reikia užpildyti kontūrais iš aibės, minimizuojant ploto sąnaudas. Taip pat duotas minimalus atstumas tarp kontūrų. Pakuojama, keičiant tarpusavio kontūrų orientaciją ir mažinant atraižų plotą.	2.10
6.	Kontūro koregavimas.	Jei kontūre trys kontroliniai taškai yra vienoje tiesėje (arba dviejose tiesėse kurios kertasi $\sim 180^\circ$ kampu), vidurinis iš jų panaikinamas. Splainai gali būti verčiami į lankus bei atkarpas pagal nurodytą slenkstinę paklaidos reikšmę.	2.11

2.3. Gaminio modelio apdorojimo literatūros analizė

Tiriama sritis konkreti ir griežtai apibrėžta, todėl galima efektyvi literatūros paieška ir analizė. Daugelyje analizuotų sričių nemažai pasiekta, todėl užtenka pasinaudoti pirmtakų patirtimi, pritaikant reikalingas modifikacijas ar papildymus.

IGES formatas. Senas (užuomazgos 1979 metais), labai mažai šaltinių. Formato standartas ir dokumentacija buvo tobulinama iki 1997 metų (*IGES v 6.0*). Beveik visos senesnės ir dauguma dabartinių sistemų naudoja šį formatą, todėl jis yra universali terpė tarp daugelio *CAD/CAM* programinių paketų. Formatas gali saugoti tik *solid* ir *NURBS* konstrukcijos tipo modelius. Šiame projekte naudojami *NURBS* kreivės ir paviršiai. Ši formatą pakeitė naujesnis *STEP* formatas (užuomazgos 1984 metais). *STEP* ir *IGES* santykis rinkoje nagrinėjamas 2.4 skyriuje.

Gaminio modelio vaizdavimas. Naudojamas plačiai paplitęs *OpenGL* vaizdavimo realiuoju laiku variklis, leidžiantis tiesiogiai vaizduoti *NURBS* kreives ir paviršius, be papildomų skaičiavimų. Gausu išsamios informacijos šaltinių.

Tūrio skaičiavimas. Užduoties nėra sudėtinga. Optimalų variantą galima rasti bandymų keliu, naudojant diskretizavimo ir integravimo metodiką. Algoritmo efektyvumas priklauso nuo geometriją aprašančios išraiškos, t.y. svarbus duomenų operatyviojoje atmintyje apie 3D paviršių saugojimo pavidalas. *NURBS* konstrukcija leidžia tiksliai apskaičiuoti paviršiaus taškus naudojant bet kokį žingsnį.

Modelio srities proporcijų keitimas. Užduotis specifinė, tačiau ją galima išskaidyti į bendrinius elementus. Konkrečią užduotį iliustruojančios medžiagos yra mažai, tačiau sprendimą galima rasti, kombinuojant sudedamųjų bendrinių elementų sprendinius. Informacijos apie bendrines (standartines) transformacijas gausu. Užduotis realizuojama, keičiant kontrolinių taškų pozicijas erdvėje, transformavimo matricų pagalba.

Dvimačių figūrų (kontūrų) pakavimas. Šioje srityje atlikta nemažai tyrimų, todėl galima rasti naudingos literatūros, nurodančios optimalius ir tikslius metodus. Rezultato kokybė (atvirkščiai proporcingas dydis sunaudotam plotui) labai priklauso nuo figūrų dėliojimo sekos. Paprastai tam naudojamas ilgio ar ploto kriterijus. Įvedus atsitiktinumo kriterijų, galima išgauti efektyvų sprendimą per užduotą laiką. Taip pat svarbus geometrinių duomenų saugojimo operatyviojoje atmintyje pavidalas (kontūro analizei reikia splainus versti atkarpomis).

Ruošdami modelį apdirbimui, *CAM* programiniai paketai analizuoja modelio kontūrą aprašančias geometrines konstrukcijas ir sudaro apdirbimo kelią, atsižvelgiant į įrankio diametrą bei kitus aparatūrinės įrangos parametrus. Šis faktas yra svarbus pakavimo uždavinio formulavime, nes

reikia atsižvelgti į fizinius įrankio matmenis, paliekant erdvės tarp pakuojamų figūrų, įrankio judesiam. Dėl tokios uždavinio specifikos dauguma siūlomų algoritmų gali išvis netikti ar būti neefektyviais. Paiešką dar susiaurina tai, kad figūros gali būti netaisyklingos, neiškilos su skylėmis viduje. Figūrų tarpusavio orientacija taip pat gali būti keičiama.

Kontūro koregavimas. Kontrolinių taškų skaičiaus minimizavimas gali būti realizuotas, sumažinant kolinearių taškų kiekį, naikinti taškus, atsižvelgiant į užduotą paklaidą. Užduotis reikalauja literatūros, aprašančios *NURBS* geometrines konstrukcijas, analizės.

Splaino vertimas į atkarpas ir lankus reikalauja papildomos informacijos, aprašančios *NURBS* geometrines konstrukcijas, analizės. Šiuolaikiniai komerciniai grafiniai paketai ne visada išveda teisingą rezultatą.

2.4. CAD/CAM rinkos analizė

Atsiradus galimybei kompiuterizuoti gamybos procesus, gaminio modelio brėžiniai iš projektuotojo lapo buvo perkelti į skaitmeninę erdvę. Pirmaujantys pasaulio gamintojai daug investavo į procesų kompiuterizavimą kaip į ateities perspektyvą. Šis žingsnis atvėrė naujas galimybes, tačiau atsirado ir ankščiau nebudingų problemų. Projektavimas tapo patogesniu, tikslesniu ir greitesniu, gamybos procesas – trumpesniu, lengviau planuojamu. Visgi išaugo šiems procesams keliami reikalavimai, buvo mesti nauji iššūkiai visiems techninės bei programinės įrangos inžinieriams ir projektuotojams.

2.4.1. Programinės įrangos tobulėjimo tendencijos

Gamybos proceso etapai ir juos realizuojantys įrankiai pateikti (žr.: *Lentelė 2.4.1*).

	Įrankis/pavidalas/etapas	Aprašymas
1.	<i>CAD</i> - gaminio modelio projektavimas	Gaminio modelis gali būti suprojektuotas, naudojant <i>CAD</i> arba <i>CAM</i> sistemą.
2.	<i>CAM</i> – apdirbimo kelių sudarymas	<i>CAM</i> sistemos įrankių pagalba generuojami apdirbimo keliai, imituojantys įrankio judėjimą.
3.	<i>NC (numeric control)</i> – apdirbimo programos generavimas	Apdirbimo keliai paverčiami <i>NC</i> programomis, kurių turinys susideda iš gamybos mašinas valdančių komandų sekos.
4.	<i>CNC (computer numeric control)</i> – apdirbimo programa valdomas gamybos procesas	<i>CNC</i> programinė įranga skaito <i>NC</i> programoje užfiksuotas komandas ir siunčia atitinkamus valdančiuosius signalus gamybos mašinai.
5.	Gaminys – gamybos proceso rezultatas	<i>NC</i> komandų sekos realizavimo rezultatas – gaminys – reali gaminio modelio kopija.

Lentelė 2.4.1: Gamybos proceso etapai, juos realizuojanti programinė įranga, gaminio aprašymo pavidalai

2005 metų rinkos statistika [6] rodo, kad sudėtingos greitaeigės penkių ašių gamybos mašinos tapo įperkamos didesniajai daliai gamintojų. Taip pat padaugėjo gamintojų, naudojančių daugiafunkcines stakles. Dėl šios priežasties išaugo sudėtingos *CAM* programinės įrangos, sugebančios generuoti, tokias gamybos mašinas valdančias programas, poreikis. Gamybos procesą aprašanti programa privalo būti optimizuota ir patikrinta, kad gaminio apdirbimas užtruktų kuo trumpiau, o programos inicijuojami veiksmai nebūtų žalingi nei gaminiui, nei gamybos mašinai.

Visus procesus stengiamasi kuo labiau automatizuoti, kad mažiau veiksmų reikėtų atlikti projektuotojui ar programuotojui. Be to, *CAM* programinė įranga, kurios pagalba kuriamos apdirbimo programos, turi būti kuo paprastesnė naudoti, o jos vartotojo sąsaja intuityviai suprantama ir personalizuojama.

Aktuali ir gaminio modelio pervedimo iš *CAD* į *CAM* sistemą problema. Nepaisant to, kad kuriamos apjungtos *CAD/CAM* sistemos, problema išlieka dėl gamintojo ir projektuotojo santykio. Dažnai gamintoją ir projektuotoją atstovauja skirtingos kompanijos, todėl projektuotojui neaktualu pirkti brangesnę programinę įrangą su aibe nereikalingų funkcijų. Projektuotojas apsiriboja *CAD* sistema ir būtinas tarpinis formatas, suprantamas tiek projektuotojo, tiek gamintojo programinėms sistemoms, kuris leistų saugiai pernešti duomenis.

2.4.2. Duomenų bylų formatai

Iki dvidešimto amžiaus aštuoniasdešimtųjų metų įvairios kompiuterizuotą gamybą vystančios kompanijos naudojo skirtingas *CAD/CAM* sistemas, skirtingas duomenų struktūras, aprašančias gaminio modelį. Duomenų mainai tarp tokių sistemų buvo neįmanomi. Ši problema buvo sprendžiama, kuriant nauja tarpinį duomenų saugojimo standartą, kurį galėtų suprasti dauguma *CAD/CAM* sistemų. 1979 metais pradėtas kurti *IGES* (žr.: 2.6 skyrius) duomenų bylos formatas. *IGES* formate aprašomos gaminių modelių geometrinės konstrukcijos gali būti skaitomos ir rašomos daugelio *CAD* bei *CAM* sistemų. 1997 metais išėjo paskutinė *IGES 6.0* formato publikacija. Formatas buvo pripažintas išsisėmusiu ir daugiau nebetobulinamas. 1984 metais, sekant *IGES* pėdomis, pradėtas kurti naujas formatas. 1994 metais išleista pirmoji *STEP* (*Standard for the Exchange of Product* – gaminio modelio aprašo formato duomenų mainams standartas – *ISO 10303*) versija. 2000 metais daugelis *CAD/CAM* sistemų jau turėjo integruotus *STEP* interpretavimo įrankius [3]. Esminis skirtumas tarp *IGES* ir *STEP* yra tas, kad *STEP* gali saugoti ne tik informaciją apie modelio geometrinę konstrukciją, bet ir papildomų duomenų, reikalingų gamybai [1]. Gaminio modelio aprašo papildymas tokiomis savybėmis suteikia galimybę aplenkti vieną kompiuterizuotos gamybos etapų – *NC* programas (žr.: Lentelė 2.4.1). Tačiau kol kas tai tik ateities vizija.

Koks *IGES* ir *STEP* santykis dabar? Vadovaujantis daugumos kompiuterinių produktų evoliucijos istorija, galima būtų teigti, jog tobulesnė versija gan greitai išstumia savo pirmtaką. Tačiau *IGES* sėkmingai naudojamas iki šių dienų, nepaisant to, kad šis standartas nebetobulinamas ir nebeatnaujinamas [3]. Rinka *STEP* priima lėčiau nei buvo tikėtasi, kas dar labiau mažina pastarojo standarto naudojimą gamyboje, turint aumenį, kad visos sistemos prieš tai sėkmingai naudojo *IGES*. Dauguma teikia pirmenybę *IGES*, kai reikia perversi netaisyklingus paviršius aprašančius duomenis iš vienos sistemos į kitą. Šiai užduočiai labai tinka *NURBS* paviršiai. Tai

turbūt vienintelis vis dar naudojamas *IGES* standarto geometrinių konstrukcijų aprašymo tipas (*solid* geometrijos standartai nebenaudojami, baigtinių elementų matricos, standumo matricos ir kt. naudojamos retai). *STEP* puikiai tinkamas *3D* brėžinių perkėlimui, tačiau *2D* perkėlimams jis nenaudojamas (nepalaikomas daugelio *CAD* sistemų). Todėl *CAD* vartotojai renkasi *IGES* ir *DXF* formatus duomenų mainams vietoje *STEP*. Nepaisant *STEP* galimybių, *CAD* įrangos gamintojai integruoja standarto interpretavimo įrankius tik geometriniams duomenims. *STEP* paplitimą taip pat stabdo prasta dokumentacija, perdėtas dvigubinimas, didelis standarto panaudojimo būdų kiekis.

Analitikai konstatuoja faktą, jog *STEP* dar nėra subrendęs ir vis dar laukia savo eilės. Skeptikai linkę manyti, kad *STEP* niekada nesugebės visiškai pakeisti *IGES*. Realus *IGES* ir *STEP* standartų privalumų palyginimas (žr.: *Lentelė 2.4.1*), naudojant skirtingus programinius paketus. *DXF* (*Drawing Exchange Format*) formatas negali būti naudojamas geometriniams *NURBS* konstrukcijoms pernešti, dėl to plačiau nenagrinėjamas.

Stand.	<i>MasterCAM v 6.0</i>		<i>MasterCAM v 9.0</i>		<i>3D Studio MAX R 5.0</i>		Palaiko <i>NURBS</i>	Palaiko <i>2D</i> geometriją	Pertekliškumas
	skaito	rašo	skaito	rašo	skaito	rašo			
<i>IGES</i>	+	+	+	+	+	+	+	+	Nedidelis
<i>STEP</i>	–	–	+	+	–	–	+	–	Didelis
<i>DXF</i>	+	+	+	+	+	+	–	+	Nedidelis

Lentelė 2.4.2: Formatų standartų, naudojamų geometrinių duomenų mainams tarp CAD/CAM sistemų, palyginimas

2.4.3. Apibendrinimas

CAD/CAM rinkos analizės metu užfiksuoti šie faktai ir tendencijos:

- Pastebimas spartus *CAD/CAM* sistemų evoliucionavimas, prisitaikant prie vartotojų poreikių. Trūksta pigios ir patogios programinės įrangos su lengvai suprantama vartotojo sąsaja.
- Patvirtintas skirtingų *CAD* ir *CAM* sistemų bylų formatų, naudojamų duomenų saugojimui, nesuderinamumas. Būtinai tarpinis formatas duomenų mainams.
- Šiuo metu pasaulyje populiarūs trys standartai, naudojami duomenim apie gaminio modelį pernešimui: *DXF*, *IGES*, *STEP*. *IGES* yra *STEP* pirmtakas, todėl apima siauresnę sferą (tik geometrinių duomenų aprašas). *STEP* standartas labai platus ir gali būti specializuotas bet kuriai gamybiniai sričiai. Tačiau *STEP* dar nėra pakankamai subrandintas ir rinka jo dar nepriima.

Atsižvelgiant į išvardintus faktus, patvirtintas kuriamos programinės įrangos reikalingumas su visomis ankščiau išvardintomis funkcijomis (žr.: 2.2 skyrius). Programos funkcijų sfera apsiriboja

geometrinių duomenų analize ir koregavimu (*CAD* funkcijų rinkinys). Geometrinei gaminio modelio konstrukcijai aprašyti pasirinktos *NURBS* konstrukcijos. Duomenų skaitymui ir rašymui naudojamas *IGES* formatas, dėl galimybės saugoti *NURBS* konstrukcijas, minimalaus pertekliško *STEP* atžvilgiu (fizinės gaminio modelio savybės ir jo apdirbimo ypatumai šiame darbe neaktualūs) ir dėl palaikymo visose nagrinėjamose *CAD/CAM* sistemose.

2.5. NURBS (neunifikuoto racionaliojo B-splaino) aprašymas

Jau 1960 metais automobilių pramonėje buvo naudojami įrenginiai galintys pasiekti vienos šimtosios milimetro tikslumą. Tačiau tada dar nebuvo būdo, leidžiančio taip tiksliai aprašyti paviršius. Net jei ir pavyktų, modelių aprašymas būtų labai grioždiškas ir neracionalus. Dėl šios priežasties buvo kuriamos naujos interpoliavimo technologijos.

Buvo sukurta daug metodų, matematiškai aprašančių kreives ir paviršius: *Bezier* kreivės, B-splainai [5]. Dėl savo lankstumo ir gebėjimo formuoti bet kokią taisyklingą ar netaisyklingą figūrą labiausiai išpopuliarėjo *NURBS* interpoliavimo metodas. Šis būdas leido bendrais bruožais aprašyti modelį, naudojant minimalų kontrolinių taškų kiekį, kai analizės tikslumas teoriškai neribotas. Modelio vaizdavimo detališkumas galėjo būti pasirinktas realiuoju laiku, nusileidžiant iki smulkiausio lygmens. To negalima buvo realizuoti su poligoniniais paviršiais ar polinominėmis kreivėmis.

Du populiariausi metodai konstruojant tikslias kreives ar paviršius – neišreikštinis (*implicit*) ir parametrinis.

Neišreikštiniame metode dažniausiai funkcija priklausanti nuo koordinačių ašių kintamųjų yra lygi 0. Taip aprašomas ryšys tarp ašių kintamųjų. Pavyzdžiui funkcija $f(x, y) = x^2 + y^2 - 1 = 0$ atvaizduoja koordinačių sistemoje xOy apskritimą su spinduliu 1.

Parametriniame metode kiekvienos ašies kintamasis yra nepriklausomo parametro funkcija. Jei u yra nepriklausomas kintamasis, tada kreivė $C(u) = [x(u), y(u)]$, $a \leq u \leq b$. Pirmąjį apskritimo koordinačių sistemoje ketvirtį galima išreikšti $C(u) = [\cos(u), \sin(u)]$, $0 \leq u \leq \pi/2$. Taip pat galima išreikšti $C(u) = [(1-t^2)/(1+t^2), 2t/(1+t^2)]$, $0 \leq t \leq 1$. Tai rodo, jog parametrinė kreivės išraiška nėra unikali.

Parametrinių kreivių (paviršių) klasei priklauso neunifikuoto racionaliojo B-splaino arba kitaip *NURBS* (*Non-uniform Rational B-Spline*) kreivė (paviršius). Šio tipo kreivės naudojamos dėl patogumo skaičiuoti kompiuteriu, stabilumo slankaus kablelio klaidų atžvilgiu, mažų atminties sąnaudų, reikalingų konstrukcijos saugojimui, ir galimybės įgyti bet kokią formą. *NURBS* yra neracionaliųjų B-splainų apibendrinimas, kurių pagrindas yra racionaliosios *Bezier* kreivės. Racionaliosios *Bezier* kreivės yra *Bezier* kreivių apibendrinimas.

n -tojo laipsnio *Bezier* kreivė apibrėžiama:

$$C(u) = \sum_{i=0}^n B_{i,n}(u)P_i, \quad 0 \leq u \leq 1 \quad (\text{Išraiška 2.5.1})$$

Geometriniai koeficientai P_i vadinami kontroliniais taškais (*Control Points*). Bazinė funkcija $B_{i,n}$ yra klasikinis n laipsnio Brenšteino polinomas:

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}. \quad (\text{Išraiška 2.5.2})$$

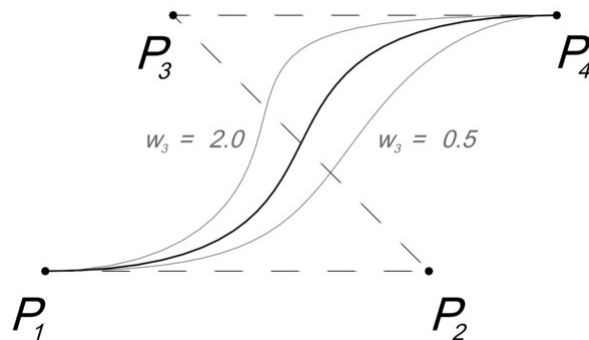
Bezier kreivės negali išreikšti antros eilės kreivių (*conic functions*). Antros eilės kreives (apskritimas, elipsė) galima išreikšti naudojant racionaliąsias funkcijas, kurios apibrėžiamos kaip dviejų polinomų santykis:

$$x(u) = \frac{X(u)}{W(u)}, \quad y(u) = \frac{Y(u)}{W(u)}, \quad z(u) = \frac{Z(u)}{W(u)}. \quad (\text{Išraiška 2.5.3})$$

Taigi racionali *Bezier* kreivė apibrėžiama kaip

$$C(u) = \frac{\sum_{i=0}^n B_{i,n}(u) w_i P_i}{\sum_{i=0}^n B_{i,n}(u) w_i}, \quad 0 \leq u \leq 1. \quad (\text{Išraiška 2.5.4})$$

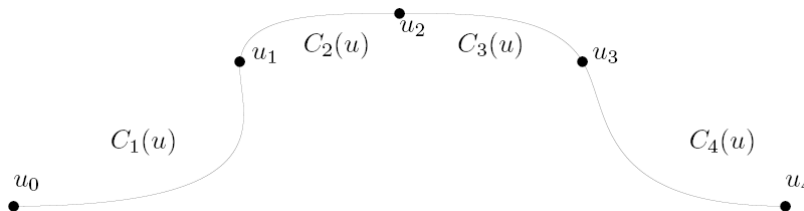
Skaliaras w_i vadinamas svoriu. Keičiant w_i , kreivė „pritraukiama“ arba „atstumama“ kontrolinio taško atžvilgiu (žr.: *Pav. 2.5.1*). Pradžioje visi svoriai lygūs 1. Didinant w_3 , kreivė priartėja prie P_3 , mažinant w_3 – kreivė nutolsta. Kai $w_3 = 0$, tada P_3 neturi jokios įtakos kreivės interpoliavimui.



*Pav. 2.5.1: racionali *Bezier* kreivė*

Parastai vienos racionaliosios *Bezier* kreivės segmento nepakanka. Keliant bazinės funkcijos laipsnį gaunama sudėtingesnė kreivė, tačiau tokią kreivę sunku efektyviai apdoroti, reikšmės skaitiškai nestabilios ir neturi galimybės lokaliai modifikuoti kreivės interaktyvaus projektavimo metu.

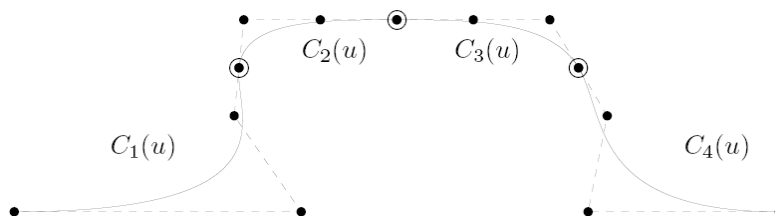
Išvengti šių problemų padeda kreivė sudaryta iš kelių racionaliųjų *Bezier* kreivių segmentų. Segmentai sujungti tam tikruose taškuose su atitinkamu tęstinumo laipsniu (*level of continuity*). Tokia kreivė vadinama B-splainu (žr.: Pav. 2.5.2).



Pav. 2.5.2: B-splainas

Kreivė $C(u)$ apibrėžta su $u \in [0; 1]$ ir susideda iš segmentų $C_i(u)$, $1 \leq i \leq m$. Segmentai sujungti taškuose $u_0 = 0 \leq u_1 \leq u_2 \leq u_3 \leq u_4 = 1$ su tam tikru tęstinumo laipsniu. Kreivė yra C_k tęstina (arba C su tęstinumo laipsniu k) taške u_i , jei $C_i^{(j)}(u_i) = C_{i+1}^{(j)}(u_i)$ su visais $0 \leq j \leq k$, kur $C_i^{(j)}$ yra j -oji C_i išvestinė.

Pav. 2.5.3 pavaizduoti B-splainą sudarantys segmentai su kontroliniais taškais. Apibraukti kontroliniai taškai (trūkio taškai – *breakpoints*) priklauso dviems *Bezier* segmentam. Suprantama, kad tokių taškų saugojimas daugiau nei vieną kartą švaisto atmintį. Jei kreivė yra C_1 tęstina, tada kai kurie kontroliniai taškai *Bezier* segmento viduje yra priklausomi nuo gretimo segmento taškų pozicijų, kad būtų patenkintas tęstinumo apribojimas. Todėl tokių taškų saugojimas nėra būtinas.



Pav. 2.5.3: B-splainą sudarantys segmentai

B-splaino išraiška turėtų efektyviai išnaudoti atmintį ir leisti lokaliai valdyti kreivę, t.y. bazinė funkcija turėtų būti apibrėžta ne $[u_0; u_m]$ intervale, tačiau naudojant baigtinį skaičių dalinių intervalų.

B-splainas apibrėžiamas taip:

$$C(u) = \sum_{i=0}^n N_{i,p} P_i, \quad a \leq u \leq b \quad (\text{Išraiška 2.5.5})$$

Čia P_i yra kontroliniai taškai, o $N_{i,p}$ yra p -tojo laipsnio bazinė B-splaino funkcija. Kompiuteriniuose skaičiavimuose naudojama rekurentinė bazinės B-splaino funkcijos išraiška.

B-splaino trūkio taškai (žr.: Pav. 2.5.2) vadinami mazgais. Tokių taškų seka vadinama mazgų vektoriumi ir apibrėžiama $U = u_0, u_1 \dots u_m$. Mazgų vektorius yra nemažėjančių realiųjų skaičių seka, t.y. $u_i \leq u_{i+1}$, $i = 0, 1 \dots m$. B-splaino p -tojo laipsnio bazinė funkcija apibrėžiama rekurentiškai:

$$N_{i,0}(u) = \begin{cases} 1, & \text{jei } u_i \leq u < u_{i+1} \\ 0, & \text{kitu atveju} \end{cases},$$

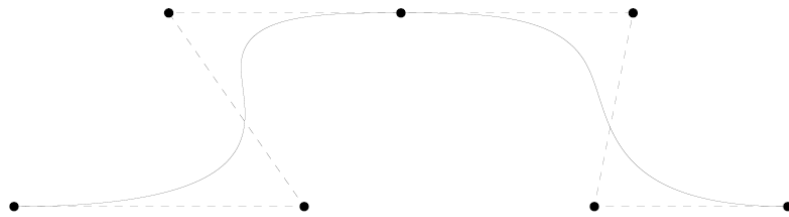
$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \quad (\text{Išraiška 2.5.6})$$

Laikoma, kad $0/0 = 0$. Mazgų vektorius yra neunifikuotas ir neperiodinis:

$$U = \{a_1 \dots a_{p+1}, u_{p+1} \dots u_{m-p+1}, b_1 \dots b_{p+1}\}. \quad (\text{Išraiška 2.5.7})$$

Čia $a_i = a_{i+1}$, $b_i = b_{i+1}$, $i = 0 \dots p-1$.

B-splainas Pav. 2.5.2 atvaizduotas Pav. 2.5.4 su jį formuojančia kontrolinių taškų lauzte. Laužtė formuojama sujungiant kontrolinius taškus P_i .



Pav. 2.5.4: B-splainas ir jį formuojančių kontrolinių taškų lauztė

Kaip buvo paminėta anksčiau, tik racionalieji splainai gali atvaizduoti antros eilės kreives. B-splaino apibendrinimas yra *NURBS*:

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u) w_i P_i}{\sum_{i=0}^n N_{i,p}(u) w_i}, \quad a \leq u \leq b. \quad (\text{Išraiška 2.5.8})$$

Čia P_i kontroliniai taškai, $N_{i,p}$ p -tojo laipsnio B-splaino bazinė funkcija, apibrėžta su neperiodiniu neunifikuotu mazgų vektorium (žr.: Išraiška 2.5.7).

Naudojant homogenines koordinates, galima racionaliąją kreivę n -matėje erdvėje išreikšti kaip polinomine kreivę $n+1$ -matėje erdvėje. Kontrolinis taškas gali būti aprašytas keturmatėje erdvėje kaip $P_i^w = w_i x_i, w_i y_i, w_i z_i, w_i$, kur $w \neq 0$. P_i gaunamas, visas koordinates padalinus iš ketvirtosios w_i . Naudojant homogenines koordinates, *NURBS* išraiška gali būti perrašyta:

$$C^w(u) = \sum_{i=0}^n N_{i,p}(u) P_i^w. \quad (\text{Išraiška 2.5.9})$$

NURBS taip pat naudojamas paviršių vaizdavimui. Toks paviršius, naudojant homogenines koordinates, išreiškiamas taip:

$$S^w(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) P_{i,j}^w. \quad (\text{Išraiška 2.5.10})$$

Čia $P_{i,j}^w$ suformuoja dvikryptį kontrolinių taškų tinklą. $N_{i,p}(u)$ ir $N_{j,q}(v)$ yra neracionaliosios bazinės funkcijos, apibrėžtos su mazgų vektoriais:

$$U = \{a_1 \dots a_{p+1}, u_{p+1} \dots u_{r-p+1}, b_1 \dots b_{p+1}\}$$

$$V = \{c_1 \dots c_{q+1}, u_{q+1} \dots u_{s-q+1}, d_1 \dots d_{q+1}\}$$

Čia $a_i = a_{i+1}$, $b_i = b_{i+1}$, $i = 0 \dots p-1$, $c_j = c_{j+1}$, $d_j = d_{j+1}$, $j = 0 \dots q-1$. Taip pat $r = n + p + 1$, $s = m + q + 1$, o ribos $[a; b]$ ir $[c; d]$ paprastai nustatomos į $[0; 1]$.

2.6. IGES bylų formato aprašymas

IGES (Initial Graphics Exchange Specification) formatas buvo pradėtas kurti 1979-aisiais metais [9]. Šio formato paskirtis – užtikrinti saugius duomenų mainus tarp skirtingų *CAD/CAM* sistemų. *IGES* yra universalus ir daug apimantis, todėl daugeliu atvejų perteklinis.

Šiame darbe nagrinėjama tik maža dalis dokumentuotų formato funkcijų, kurių pakaks užduočiai įvykdyti.

2.6.1. IGES struktūra

Byla gali būti sugeneruota naudojant *ASCII* ar dvejetainius formatus. Taip pat duomenys gali būti glaudinti. Dvejetainio formato ir glaudinti duomenys šiame darbe nenagrinėjami.

Visas bylos turinys padalintas į penkias sekcijas: *S (Start)* - pradžios, *G (Global)* - globalioji, *D (Directory Entry)* – adresų įrašų, *P (Parameter Data)* – parametrų, *T (Terminate)* -pabaigos. Kiekviena sekcija turi griežtai apibrėžtą struktūrą ir paskirtį:

- *S* – nėra būtina, joje gali būti įrašyti programinės įrangos, sugeneravusios bylą komentarai ar vartotojo tekstas.
- *G* – saugomi globalūs bylos parametrai, tokie kaip: kūrimo data, autorius, išėties bylos kelias, PĮ (sugeneravusi bylą), skirtukai naudojami byloje ir pan.
- *D* – atlieka esybių, aprašytų byloje, turinio funkciją. Taip pat šioje sekcijoje pateikiamos kelios parametrų reikšmės, nesusijusios su geometrine esybių erdve.
- *P* – patalpinti visi esybę aprašantys parametrai (koordinatės, tipų reikšmės, rodyklės).
- *T* – būtina sekcija, deklaruojanti bylos pabaigą, joje pažymėtos visų kitų sekcijų eilučių sumos.

Visos neglaudinto *ASCII* formato eilutės yra 80 simbolių ilgio (žr.: *Lentelė 2.6.1*) . *S* sekcija nebūtina, *G* sekcija paprastai užima 2 ar daugiau eilučių. *T* turi tik vieną eilutę. *D* ir *P* eilučių skaičius priklauso nuo aprašytų esybių kiekio, kadangi *D* sekcijoje vienai esybei aprašyti išskirtos 2 eilutės – jų skaičius visada lyginis.

byla sudaroma ir nuskaitoma, vadovaujantis taisyklėmis:

1. Sekcijų eiliškumas yra griežtai apibrėžtas ir negali kisti: *S, G, D, P, T*.
2. Sekciją žymi raidė esanti kiekvienos eilutės 73-iame stulpelyje.
3. Sekcijos turi savo eilučių numeravimą nuo 1. Numeriai pateikiami dešimtainiu kodu 74 – 80 bylos stulpeliuose, tušti stulpeliai gali būti užpildyti nuliais.
4. Byloje negali būti tuščių eilučių, o visa informacija esanti žemiau *T* sekcijos ignoruojama.

	0000000001111111112222222222333333333344444444445555555555666666666677777777778											
	1234567890123456789012345678901234567890123456789012345678901234567890											
...	...											
007	143	1	0	1	0	0	0	0	00000000D	1		
008	143	0	-19	1	0	0	0	0	0D	2		
009	128	2	0	0	0	0	0	0	00001000D	3		
010	128	0	0	78	0	0	0	0	0D	4		
011	141	80	0	0	0	0	0	0	00001000D	5		
012	141	0	0	1	0	0	0	0	0D	6		
013	126	81	0	0	0	0	0	0	00001000D	7		
014	126	0	0	13	0	0	0	0	0D	8		
015	126	94	0	0	0	0	0	0	00001050D	9		
016	126	0	0	3	1	0	0	0	0D	10		
017	126	97	0	0	0	0	0	0	00001000D	11		
018	126	0	0	14	0	0	0	0	0D	12		
019	126	111	0	0	0	0	0	0	00001050D	13		
020	126	0	0	3	1	0	0	0	0D	14		
021	126	114	0	0	0	0	0	0	00001000D	15		
022	126	0	0	13	0	0	0	0	0D	16		
023	126	127	0	0	0	0	0	0	00001050D	17		
024	126	0	0	3	1	0	0	0	0D	18		
025	314	130	0	0	0	0	0	0	00000020D	19		
026	314	0	6	1	0	0	0	0	0D	20		
027	143,1,3,1,5;									1P	1	
028	128,7,7,3,3,0,1,1,0,0,0.0D0,0.0D0,0.0D0,0.0D0,									3P	2	
029	6.61209583980858D0,13.469646436108D0,18.2476946095025D0,									3P	3	
030	24.4128473147503D0,30.1247473123735D0,30.1247473123735D0,									3P	4	
031	30.1247473123735D0,30.1247473123735D0,0.0D0,0.0D0,0.0D0,0.0D0,									3P	5	
032	5.62306912373312D0,11.1126212298475D0,16.7202631492319D0,									3P	6	
033	22.2098151827482D0,27.8328843454078D0,27.8328843454078D0,									3P	7	
034	27.8328843454078D0,27.8328843454078D0,1.0D0,1.0D0,1.0D0,1.0D0,									3P	8	
035	1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,									3P	9	
...	...											
040	1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,									3P	14	
041	-13.8753623962402D0,2.83775447871171D-6,-113.599208831787D0,									3P	15	
042	-59.552116394043D0,2.14153025766831D-6,-114.44507598877D0,									3P	16	
043	-54.263162612915D0,3.60738415278321D-6,-82.5273284912109D0,									3P	17	
...	...											
104	-15.8671588897705D0,-2.75817247332558D-6,63.0996322631836D0,									3P	78	
105	0.0D0,30.1247473123735D0,0.0D0,27.8328843454078D0;									3P	79	
106	141,1,1,3,3,7,1,1,9,11,1,1,13,15,2,1,17;									5P	80	
...	...											
120	126,1,1,0,0,1,0,0.0D0,0.0D0,1.0D0,1.0D0,1.0D0,1.0D0,									9P	94	
121	30.1247473123735D0,27.8328843454078D0,0.0D0,0.0D0,									9P	95	
122	27.8328843454078D0,0.0D0,0.0D0,1.0D0,0.0D0,0.0D0,0.0D0;									9P	96	
123	126,7,3,1,1,1,0,0.0D0,0.0D0,0.0D0,0.0D0,5.62306912373312D0,									11P	97	
...	...											
135	63.0996322631836D0,0.0D0,27.8328843454078D0,0.0D0,0.0D0,									11P	109	
136	-.99999999999993D0;									11P	110	
137	126,1,1,0,0,1,0,0.0D0,0.0D0,1.0D0,1.0D0,1.0D0,1.0D0,									13P	111	
138	30.1247473123735D0,0.0D0,0.0D0,30.1247473123735D0,									13P	112	
139	27.8328843454078D0,0.0D0,0.0D0,1.0D0,0.0D0,0.0D0,0.0D0;									13P	113	
140	126,7,3,0,0,1,0,0.0D0,0.0D0,0.0D0,0.0D0,6.61209583980858D0,									15P	114	
...	...											
152	63.0996322631836D0,0.0D0,30.1247473123735D0,0.0D0,0.0D0,0.0D0;									15P	126	
153	126,1,1,0,0,1,0,0.0D0,0.0D0,1.0D0,1.0D0,1.0D0,1.0D0,									17P	127	
154	30.1247473123735D0,27.8328843454078D0,0.0D0,0.0D0,									17P	128	
155	27.8328843454078D0,0.0D0,0.0D0,1.0D0,0.0D0,0.0D0,0.0D0;									17P	129	
156	314,65.0D0,0.0D0,65.0D0,;									19P	130	
157	S	1G	5D	20P	130				T	1		

Lentelė 2.6.1: IGES bylos pavyzdys, skaičiai žaliai fone neįeina į bylos turinį, jie žymi eilučių ir stulpelių numerius

2.6.2. Pradžios sekcija S

Sekcija skirta komentarams ir nėra privaloma, gali būti visiškai tuščia, tokiu atveju užpildyti tik 73 - 80 stulpeliai.

2.6.3. Globalioji sekcija G

Čia saugomi personaliniai duomenys, sugeneruoti specifinės programinės įrangos, toki kaip išeities bylos loginis kelias, bylos autorius, programinė įranga, kuri sugeneravo bylą, kūrimo data bei laikas ir panašiai (žr.: Lentelė 2.6.2).

	00000000011111111122222222233333333344444444455555555666666667777777778 12345678901234567890123456789012345678901234567890123456789012345678901234567890
...	...
002	1H, 1H; 23HMASTERCAM version 6.13a, 12HMG_TEST0.ge3, 9HMASTERCAM, 1H1, 16, 8, G0000001
003	24, 8, 56, , 1, 2, 2HMM, 1, 0.01, 13H060328.131224, 0.001, 100., , , 8, 0, ; G0000002
...	...

Lentelė 2.6.2: IGES bylos G sekcijos pavyzdys, pilkai pažymėtos svarbių parametru reikšmės

Svarbūs parametrai yra: skirtukai (simboliai, atskiriantys parametru reikšmes ar grupes), modelio erdvės mastelis, vienetai, minimali rezoliucija, apytikslė maksimali koordinatės reikšmė.

Kiekvienas parametras gali būti sveikasis skaičius, realusis skaičius arba simbolių seka. Prieš kiekvieną simbolių seką yra sveikasis skaičius nurodantis sekos ilgį ir atskirtas simboliu „H“ nuo pačios sekos (pvz.: „4HINCH“ žymi simbolių seką „INCH“ iš 4 simbolių).

Kadangi parametru eiliškumas yra fiksuotas, nesvarbius parametrus galima praleisti, skaičiuojant skirtukus (žr.: Lentelė 2.6.3).

Parametro eilės numeris	Parametro reikšmė ir komentarai
1	Parametru reikšmių skirtukas (Pvz. „1H,“ reikškia kad parametru reikšmių atskyrimui bus naudojamas kablelis). Standartiškai naudojamas kablelis.
2	Įrašų skirtukas. Standartiškai naudojamas kabliataškis.
13	Modelio erdvės mastelis. Standartiškai 1.
14	Vienetų tipą žyminti vėliavėlė: 1 – coliai, 2 – milimetrai, 4 – pėdos, 5 – mylios, 6 – metrai, 7 – kilometrai, 9 – mikronai, 10 – centimetrai. Jei vėliavėlės reikšmė 3 – vienetų pavadinimas gali būti nuskaitomas iš 15 parametro. Standartiškai 1.
19	Minimalus vartotojo ar programinės įrangos užduotas tikslumas.
20	Apytikslė maksimali koordinatės reikšmė, naudota aprašant esybes. Naudinga, užduodant scenos ribas.

Lentelė 2.6.3: G sektoriaus parametru reikšmės

2.6.4. Adresų įrašų sekcija D

Adresų įrašų sekcija atlieka turinio, byloje aprašytoms esybėms, funkciją. Kiekvienas įrašas yra fiksuoto formato: išdėstytas dviejose gretimose eilutėse, turi 20 parametrų, kurių reikšmėms skirta po 8 simbolius. Parametrų reikšmės lygiuojamos ties dešiniu kraštu (žr.: *Lentelė 2.6.4*).

	0000000001111111112222222222333333333344444444445555555555666666666677777777778																				
	1234567890123456789012345678901234567890123456789012345678901234567890																				
...	...																				
007	143	1	0	1	0	0	0	0	000000000D	1											
008	143	0	-19	1	0	0	0	0	0D	2											
009	128	2	0	0	0	0	0	0	000010000D	3											
010	128	0	0	78	0	0	0	0	0D	4											
011	141	80	0	0	0	0	0	0	000010000D	5											
012	141	0	0	1	0	0	0	0	0D	6											
013	126	81	0	0	0	0	0	0	000010000D	7											
014	126	0	0	13	0	0	0	0	0D	8											
...	...																				
025	314	130	0	0	0	0	0	0	000000200D	19											
026	314	0	6	1	0	0	0	0	0D	20											

Lentelė 2.6.4: IGES bylos D sekcijos pavyzdys

Visi parametrai, išskyrus 10, 16, 17, 18, 20, yra sveikieji skaičiai arba rodyklės. Visoms neužpildytoms (tuščioms) parametrų reikšmėms (išskyrus 1, 2, 10, 11, 14, 20) bus priskirtos standartinės reikšmės pagal nutylėjimą. Paprastai tušties laukams generatorius priskiria reikšmę „0“, o nuskaitanti programa reikšmę ignoruoja.

Šioje sekcijoje teigiami sveikieji skaičiai atitinka parametrų reikšmes, o neigiamų skaičių absoliutinės reikšmės atitinka rodyklės į kitą įrašą sekcijoje D, kur gali būti saugoma daugiau parametrų reikšmių.

Kiekvienas parametras turi specifinę paskirtį ir jo reikšmė gali būti interpretuojama skirtingai nuo kitų (žr.: *Lentelė 2.6.5*).

1 - 8	9 - 16	17 - 24	25 - 32	33 - 40	41 - 48	49 - 56	57 - 64	65 - 72	73	74 - 80
Esybės tipo numeris #	Parametrų duomenys #	Struktūra #, =>	Linijos piešimo tipas #, =>	Lygis #, =>	Vaizdas 0, =>	Transformac. matrica 0, =>	Įvardinimo vaizd. asociacija 0, =>	Būsenos numeris #	D	Eilės numeris #, =>
Esybės tipo numeris #	Linijos storio numeris #	Spalvos numeris #, =>	Parametrų eilučių kiekis #	Formos numeris #	Rezervuota	Rezervuota	Esybės įvardinimas	Esybės scenarijaus numeris #	D	Eilės numeris #+1

Lentelė 2.6.5: IGES bylos D sekcijos įrašo formatas, # - sveikas sk., => - rodyklė, 0 - nulis arba rodyklė. Žaliai pažymėti nagrinėtini įrašo laukai

- **Esybės tipo numeris (1, 11)** – apibrėžia esybės tipą.
- **Parametrų duomenys (2)** – pirmos esybės aprašančios eilutės numeris parametrų sekcijoje.

- **Transformacijų matrica (7)** – rodyklė į transformacijų matricos įrašą adresu sekcijoje, jei esybei taikomos transformacijos arba „0“.
- **Būsenos numeris (9)** – apibrėžia esybės būseną pagal 4 aspektus, kiekvienam naudojant po 2 baitus:
 - a. 1, 2 baitai (*matomumas*): „00“ – matoma, „01“ – nematoma.
 - b. 3,4 baitai (*priklausomybė*): „00“ – nepriklausoma, „01“ – fiziškai priklausoma, „02“ – logiškai priklausoma, „03“ – priklausoma tiek fiziškai tiek logiškai
 - c. 5,6 baitai (*esybės paskirties vėliavėlė*): „00“ – geometrija, „01“ – anotacija, „02“ – apibūdinimas, „03“ – kita, „04“ – loginė/pozicinė, „05“ – 2D parametrinė, konstrukcinė geometrija
 - d. 7,8 baitai (*hierarchijos tipas*): „00“ – globali iš viršaus žemyn, „01“ – globali priklausomybė, „02“ – naudoja apibrėžtas hierarchijos savybes.
- **Eilės numeris (10, 20)** – sekcijos D eilutės numeris.
- **Spalvos numeris (13)** – spalvos numeris arba rodyklė į kitą įrašą adresu sekcijoje, aprašantį spalvą. Yra 8 standartinių spalvų numeriai (žr.: *Lentelė 2.6.6*).
- **Parametrų eilučių kiekis (14)** – parametrų sekcijos eilučių kiekis, kuriose išvardintos esybės parametrų reikšmės.
- **Formos numeris (15)** – numeris, nurodantis kaip konkretus aprašomos esybės tipas bus interpretuojamas, kiekvienam esybių tipui gali būti būdingos kelios variacijos.

Spalvos numeris	Spalvos pavadinimas
0	Jokios spalvos (standartiškai)
1	Juoda
2	Raudona
3	Žalia
4	Mėlyna
5	Geltona
6	Rausvai purpurinė
7	Žydra
8	Balta

Lentelė 2.6.6: standartinių spalvų numeriai

2.6.5. Parametrų sekcija P

Čia saugomi visi duomenys, aprašantys esybes. Sekcijos formatas nėra taip griežtai apibrėžtas, kaip adresų įrašų. Parametrų reikšmės rašomos paeiliui ir atskiriamos skirtukais. Reikšmių formatas gali būti įvairus (žr.: *Lentelė 2.6.7*).

	0000000001111111112222222222333333333344444444445555555555666666666677777777778		
	12345678901234567890123456789012345678901234567890123456789012345678901234567890		
...	...		
027	143, 1, 3, 1, 5;	1P	1
028	128, 7, 7, 3, 3, 0, 1, 1, 0, 0, 0.0D0, 0.0D0, 0.0D0, 0.0D0,	3P	2
029	6.61209583980858D0, 13.469646436108D0, 18.2476946095025D0,	3P	3
030	24.4128473147503D0, 30.1247473123735D0, 30.1247473123735D0,	3P	4
031	30.1247473123735D0, 30.1247473123735D0, 0.0D0, 0.0D0, 0.0D0, 0.0D0,	3P	5
032	5.62306912373312D0, 11.1126212298475D0, 16.7202631492319D0,	3P	6
033	22.2098151827482D0, 27.8328843454078D0, 27.8328843454078D0,	3P	7
034	27.8328843454078D0, 27.8328843454078D0, 1.0D0, 1.0D0, 1.0D0, 1.0D0,	3P	8
035	1.0D0, 1.0D0, 1.0D0, 1.0D0, 1.0D0, 1.0D0, 1.0D0, 1.0D0, 1.0D0,	3P	9
...	...		
104	-15.8671588897705D0, -2.75817247332558D-6, 63.0996322631836D0,	3P	78
105	0.0D0, 30.1247473123735D0, 0.0D0, 27.8328843454078D0;	3P	79
...	...		

Lentelė 2.6.7: IGES bylos P sekcijos pavyzdys

Kiekvienam esybės tipui būdingas savitas parametrų išdėstymas ir formatas, tačiau egzistuoja bendros taisyklės sekcijos sudarymui.

Pirmoji reikšmė visada nurodo esybės tipo numerį. Visos parametrų reikšmės laisvu formatu išdėstytos stulpeliuose nuo 1 iki 64, stulpelyje 65 yra tarpo simbolis. Nuo 66 iki 72 stulpeliuose fiksuotu formatu pateikta esybių numeracija pagal adresų sekcijos eilutės numerį (pvz.: „128“ tipo esybės iš lentelės .1b 9 eilutės ir 74 – 80 stulpelių reikšmė atitinka reikšmes iš .1c lentelės 28 – 105 eilučių ir 66 – 72 stulpelių). 73 stulpelyje simbolis „P“, o stulpeliuose 74 – 80 žymimas sekcijos eilutės numeris.

Konkretūs esybių tipai ir jų parametrų prasmė patekti *Prieduose 8.1* skyriuje.

2.6.6. Pabaigos sekcija T

Ši sekcija turi vienintelę eilutę, kurios 73 – 80 stulpeliai užpildyti, vadovaujantis ankščiau minėtomis taisyklėmis: „T0000001“ arba „T 1“ (žr.: *Lentelė 2.6.8*).

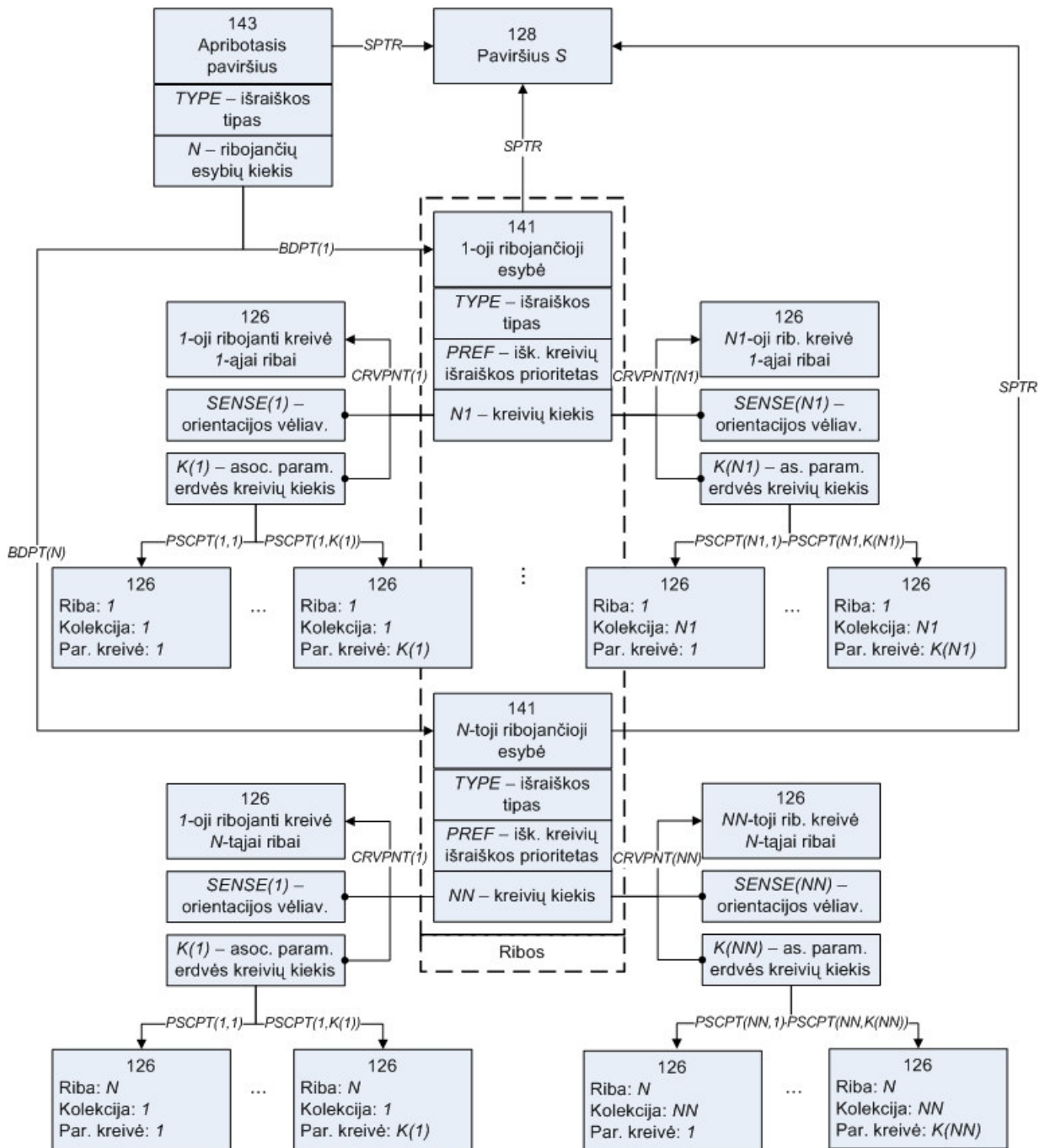
	0000000001111111112222222222333333333344444444445555555555666666666677777777778						
	12345678901234567890123456789012345678901234567890123456789012345678901234567890						
...	...						
157	S	1G	5D	20P	130	T	1

Lentelė 2.6.8: IGES bylos T sekcijos pavyzdys

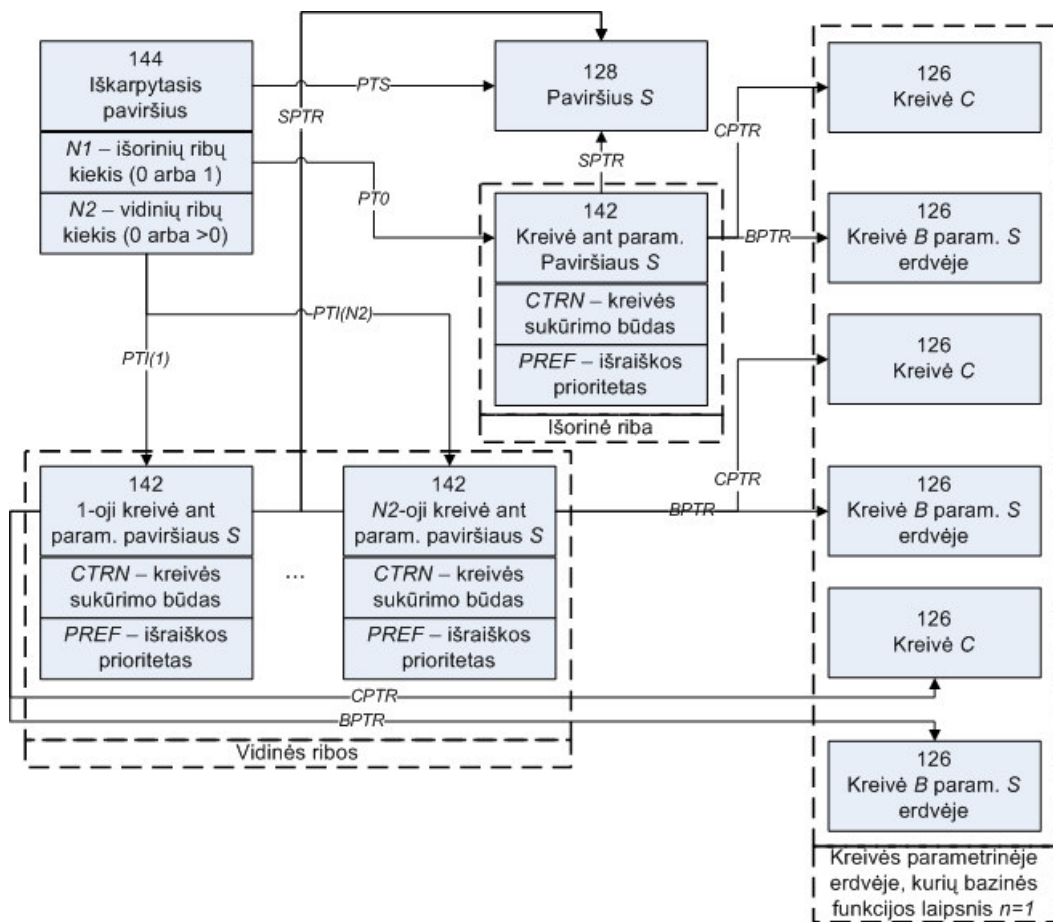
Pabaigos sekcijoje saugomi kitų sekcijų eilučių kiekiai: Nuo 02 iki 08 baido – S sekcijos, nuo 10 iki 16 baido – G sekcijos, nuo 18 iki 23 baido – D sekcijos, nuo 25 iki 31 baido – P sekcijos.

2.6.7. Apibendrinimas

Išstudijavus visus nagrinėtinius tipus, aprašytus 8.1 skyriuje, buvo sudarytos apytikslės IGES bylos interpretavimo schemas. Nagrinėjamas apkarpytasis NURBS paviršius, nes ši esybė naudoja beveik visus nagrinėtinius esybių tipus. Eksperimentai su konkrečiomis programinėmis įrangomis (*MasterCAM* ir *3D Studio Max*) parodė, kad skirtingos sistemos skirtingai interpretuoja IGES standartą. Patikslintos interpretavimo schemas, aprašančios iškarpytąjį paviršių, pateiktos (žr.: Pav. 2.6.1, Pav. 2.6.2).



Pav. 2.6.1: IGES standarto schema apribotajam paviršiui, naudojama 3D Studio Max 5.0 sistemoje



Pav. 2.6.2: IGES standarto schema apribotajam paviršiui, naudojama MasterCAM v 6.0 ir vėlesnėse sistemos versijose

Kaip matyti iš schemų, MasterCAM sistema naudoja visiškai kitus esybių tipus (144, 142 – iškarpytojo paviršiaus struktūros schema), kad aprašyti iškarpytojo paviršiaus struktūrą, nei 3D Studio Max (143, 141 – apribotojo paviršiaus struktūros schema). Struktūros hierarchijos apačioje esantys baziniai esybių tipai (128, 126) sutampa. Antrojo hierarchijos lygmens esybės (142, 141) abiejose sistemose rodo į dviejų rūšių iškarpančiąsias/ribojančiąsias kreives (126 tipas): modelio erdvėje ir parametrų erdvėje. Pagal OpenGL aprašymą (žr.: 2.7.3 skyrelis) iškarpymui naudojamos kreivės parametrinėje erdvėje. Todėl paviršius (128 tipas), į kurį rodo apribotojo/iškarpytojo paviršiaus esybė (143/144 tipas), iškarpomas, naudojant kreives (126 tipas), į kurias rodo $PSCPT/BPTR$ rodyklės.

2.7. OpenGL NURBS aprašymas

Žemiausiame aparatūriniame lygmenyje geometrinių objektų vaizdavimui naudojamos atkarpos, trikampiai ar keturkampiai (sudaryti iš dviejų trikampių) [7][10]. Glotniems paviršiams ar kreivėms vaizduoti, naudojamos aproksimacijos. Kuo didesnis trikampių, aproksimuojančių paviršių skaičius, tuo tikslesnis gaunamas paviršiaus vaizdas. Tokio paviršiaus struktūrą aprašo keletas kontrolinių taškų, mazgų vektorius ir taškų svoriai. Tarpinės reikšmės apskaičiuojamos naudojant bazines B-splaino funkcijas, kontrolinių taškų pozicijas, mazgų vektorių, svorių reikšmes ir užduotą tikslumą.

OpenGL leidžia vaizduoti *NURBS* paviršius ir kreives naudojant *evaluator'ius* (žemesnio lygmens komandos) arba *GLU* bibliotekos įrankius (aukštesniojo lygmens komandos). Aukštesniojo lygio komandos naudoja žemesniojo lygmens komandas sudėtingesnėm ir didesnėm apimties užduotim realizuoti. Taip taupomas laikas ir programos teksto kodas. Todėl šiame skyriuje nagrinėjami tik *GLU* įrankiai.

Šiame skyriuje nenagrinėjami *OpenGL* pagrindai.

Detalų *NURBS* aprašymą galima rasti 2.5 skyriuje.

2.7.1. NURBS kreivė

NURBS kreivės vaizdavimas naudojant *GLU* įrankius yra labai panašus į *NURBS* paviršiaus vaizdavimą. Skirtumas tas, kad kreivės apskaičiavimui naudojamas vienas parametras u , o paviršiaus apskaičiavimui naudojami du – u ir v . Taip pat skiriasi tam tikros procedūros ir argumentai.

NURBS kreivės aprašas turi prasidėti *gluBeginCurve(GLUnurbsObj *nobj)* ir pasibaigti *gluEndCurve(GLUnurbsObj *nobj)* procedūromis. Tarp šių procedūrų aprašoma visa kreivės konstrukcija ir vaizdavimo būdas. Kreivės atributai ir konstrukcija nustatomi naudojant *gluNurbsCurve()*. Vienas iš *gluNurbsCurve()* parametrų privalo apibrėžti kreivės taškų generavimo tipą: *GL_MAPI_VERTEX_3* arba *GL_MAPI_VERTEX_4*. Pirmasis pasirinkimas aktyvuoja taškų generavimą, naudojant įprastines 3D koordinates (x, y, z), antrasis – homogenines koordinates (x, y, z, w). Pastarosios paprastai aktyvuojamos dirbant su racionaliosiomis kreivėm (kada kontrolinių taškų svoriai nėra vienodi).

Žemiau pateiktos *gluNurbsCurve()* parametrų reikšmės:

```
void gluNurbsCurve (GLUnurbsObj *nobj, GLint uknot_count,  
GLfloat *uknot, GLint u_stride, GLfloat *ctlarray,  
GLint uorder, GLenum type);
```


- **nobj* – rodyklė į NURBS vaizduojamą objektą.
- *uknot_count* – mazgų vektoriaus ilgis.
- **uknot* – rodyklė į mazgų vektorių.
- *u_stride* – kintamųjų kiekis reikalingas aprašyti kreivę tarp dviejų kontrolinių taškų. Jis priklauso nuo *uorder* ir *type*. Pavyzdžiui (žr.: 2.7.3 skyrelio pavyzdys) iškarpančiai kreivei $u_stride = 2$ (x ir y koordinatinių reikšmės).
- **ctlarray* – rodyklė į kontrolinių taškų vektorių (kiekvienas vektoriaus elementas taip pat vektorius su x , y , z koordinatinių reikšmėmis).
- *uorder* – splaino su u parametru eilė (bazinės funkcijos laipsnis + 1).
- *type* – taškų generavimo tipas.

Kontrolinių taškų skaičius lygus mazgų vektoriaus ilgiui minus u parametru eilė.

2.7.2. NURBS paviršius

NURBS paviršiaus aprašas turi prasidėti *gluBeginSurface()* ir pasibaigti *gluEndSurface()* procedūromis. Paviršiaus atributai ir konstrukcija nustatomi naudojant *gluNurbsSurface()*. Paviršius generuojamas dviejų kreivių: viena eina išilgai paviršiaus (parametras u), kita – skersai (parametras v).

```
void gluNurbsSurface (GLUnurbsObj *nobj, GLint uknot_count,
GLfloat *uknot, GLint vknot_count, GLfloat *vknot,
GLint u_stride, GLint v_stride, GLfloat *ctlarray,
GLint uorder, GLint vorder, GLenum type);
```

- **nobj* – rodyklė į NURBS vaizduojamą objektą.
- *uknot_count* – mazgų vektoriaus ilgis generuojančiai kreivei su parametru u .
- **uknot* – rodyklė į mazgų vektorių generuojančiai kreivei su parametru u .
- *vknot_count* – mazgų vektoriaus ilgis generuojančiai kreivei su parametru v .
- **vknot* – rodyklė į mazgų vektorių generuojančiai kreivei su parametru v .
- *u_stride* – kintamųjų kiekis reikalingas aprašyti paviršių tarp dviejų kontrolinių taškų u parametru kryptimi. Pavyzdžiui (žr.: 2.7.3 skyrelio pavyzdys) $u_stride = 12 = 3*4$, jei $type = GL_MAPI_VERTEX_3$ – naudojamos 3 koordinatės taškų pozicijom saugoti (x , y ir z) bei $vknot_count - vorder = 4$ (4 kontroliniai taškai v parametru kryptimi).
- *v_stride* – kintamųjų kiekis reikalingas aprašyti paviršių tarp dviejų kontrolinių taškų v parametru kryptimi. Anksčiau minėtame pavyzdyje $v_stride = 3$, nes

naudojamos 3 koordinatės, o kontroliniai taškai v parametro kryptimi laikomi kaimyniniais vienas kito atžvilgiu.

- **ctlarray* – rodyklė į kontrolinių taškų matricą.
- *uorder* – splaino su u parametru eilė (bazinės funkcijos laipsnis + 1).
- *vorder* – splaino su v parametru eilė (bazinės funkcijos laipsnis + 1).
- *type* – taškų generavimo tipas.

Kontrolinių taškų u ir v parametrų kryptimis nustatomas pagal atitinkama mazgų vektorius ilgį, atėmus splaino su atitinkamu parametru eilę.

2.7.3. Iškarpytasis NURBS paviršius

Kad atvaizduoti iškarpytą paviršių, pirmiausiai reikia pasiruošti neiškarpyta paviršių, kaip aprašyta 2.7.2 skyrelyje. Tarp *gluBeginSurface()* ir *gluEndSurface()* turi būti *gluNurbsSurface()*. Po paviršiaus aprašo, prieš *gluEndSurface()* turi būti *gluBeginTrim()* ir *gluEndTrim()* procedūros. Tarp pastarųjų privalo būti objektai, aprašantys iškarpančiąją kreivę. Tokiais objektais gali būti laužtė *gluPwlCurve()* arba NURBS kreivė *gluNurbsCurve()*. Šie objektai turi sudaryti orientuotą uždara kreivę, kuri vadinama iškarpančiąja.

Iškarpančioji kreivė privalo neišeiti iš parametrinės (u, v) erdvės ribų, viskas kreivei iš dešinės bus iškirpta, o viskas iš kairės – vaizduojama. Dėl šios priežasties teisinga išorinių ir vidinių kreivių orientacija labai svarbi. Išorinės iškarpančiosios kreivės kryptis turi atitikti kryptį prieš laikrodžio rodyklę. Iškarpančiosios kreivės negali kirstis tarpusavyje.

Iškarpančiosios kreivės kūrimui *gluNurbsCurve()* pagalba naudojami *GLU_MAP1_TRIM_2* (2D koordinatės – x ir y) ir *GLU_MAP1_TRIM_3* (homogeninės 2D koordinatės – x, y, w) taškų generavimo tipai. Todėl *u_stride* reikšmė lygi 2 arba 3.

Iškarpytojo paviršiaus programos teksto pavyzdys:

```
void display(void)
{
    GLfloat knots[8] = {0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0};
    GLfloat edgePt[5][2] = /* counter clockwise */
    {{0.0, 0.0}, {1.0, 0.0}, {1.0, 1.0}, {0.0, 1.0}, {0.0, 0.0}};
    GLfloat curvePt[4][2] = /* clockwise */
    {{0.25, 0.5}, {0.25, 0.75}, {0.75, 0.75}, {0.75, 0.5}};
    GLfloat curveKnots[8] =
    {0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0};
    GLfloat pwlPt[4][2] = /* clockwise */
    {{0.75, 0.5}, {0.5, 0.25}, {0.25, 0.5}};
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glPushMatrix();
    glRotatef(330.0, 1., 0., 0.);
    glScalef(0.5, 0.5, 0.5);
    gluBeginSurface(theNurb);
```

```

gluNurbsSurface(theNurb, 8, knots, 8, knots,
4 * 3, 3, &ctlpoints[0][0][0], 4, 4, GL_MAP2_VERTEX_3);
gluBeginTrim (theNurb);
gluPwlCurve (theNurb, 5, &edgePt[0][0], 2, GLU_MAP1_TRIM_2);
gluEndTrim (theNurb);
gluBeginTrim (theNurb);
gluNurbsCurve (theNurb, 8, curveKnots, 2,
&curvePt[0][0], 4, GLU_MAP1_TRIM_2);
gluPwlCurve (theNurb, 3, &pwlPt[0][0], 2, GLU_MAP1_TRIM_2);
gluEndTrim (theNurb);
gluEndSurface(theNurb);
glPopMatrix();
glFlush();
}

```

Šiame pavyzdyje pirmo iškarpančioji kreivė yra išorinė ir eina parametru erdvės riba prieš laikrodžio rodyklė. Tai užtikrina, kad bus vaizduojamas visas paviršius be iškarpymų. Antroji iškarpančioji kreivė sudaryti iš *NURBS* kreivės ir laužtės bei atitinka laikrodžio rodyklės kryptį.

2.8. Tūrio skaičiavimo algoritmas

Uždavinys: Paviršiai $S_1(u, v)$, $S_2(u, v) \dots S_n(u, v)$ ir pagalbinės plokštumos $a_x, b_x, a_y, b_y, a_z, b_z$ naudojamos uždarei trimatės erdvės sričiai nustatyti, kurios tūrį reikia skaičiuoti. Tūris skaičiuojamas, atsižvelgiant į diskretizavimo žingsnio reikšmę d .

Analizė: Tarkime, kad paviršiai $S_k(u, v)$, $k = 1, 2 \dots n$ yra nepersidengiantys. Be to, paviršiams galioja apribojimas:

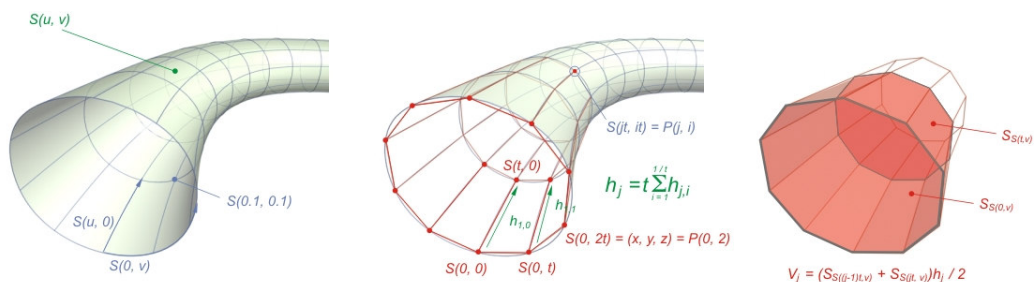
$$S_j(0, v) = S_{j-1}(1, v). \quad (\text{Išraiška 2.8.1})$$

Čia $u \in [0; 1]$, $v \in [0; 1]$, $j = 2, 3 \dots n$, o paviršiai jungiami u parametro kryptimi. Kitaip sakant, gretimų paviršių galai sutampa. Jei paviršių rinkinys nesuformuoja uždaros erdvės – būtinai naudojamos pagalbinės plokštumos.

Jei erdvė apribota, laikantis šių reikalavimų, tada bendras erdvės tūris lygus visų paviršių tūrių sumai:

$$V = \sum_{k=1}^n V_k. \quad (\text{Išraiška 2.8.2})$$

Tūris V_k skaičiuojamas kiekvienam paviršiui $S_k(u, v)$, jį diskretizuojant pasirinktu žingsniu t , išilgai parametro u ir parametro v (žr.: Pav. 2.8.1). Žingsnis t pasirenkamas atsižvelgiant į d reikšmę.



Pav. 2.8.1: Paviršiaus tūrio skaičiavimas

Idealiu atveju paviršiui išilgai parametro v galioja savybė:

$$S(u, 0) = S(u, 1). \quad (\text{Išraiška 2.8.3})$$

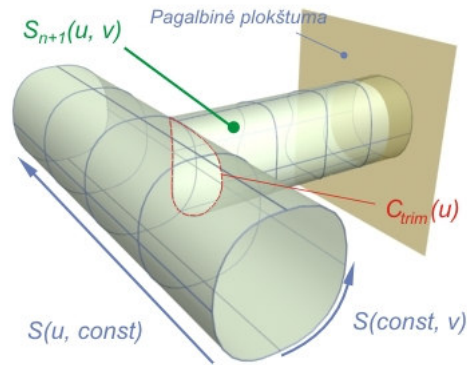
Tai reiškia, kad kreivės, einančios išilgai parametro v galai sutampa (kreivė uždara) ir jos kelyje nėra nė vienos pagalbinės plokštumos. Galimi kiti atvejai:

Išimtis 1. Kreivės, einančios išilgai parametro v , galai nesutampa, tačiau atvirą erdvę riboja viena ar kelios pagalbinės plokštumos.

Išimtis 2. Kreivės, einančios išilgai parametro v , dalis (ar kelios dalys) yra atkirstos pagalbinių plokštumų.

Išimtis 3. Kreivės, einančios išilgai parametro v , kelyje yra iškarpančioji kreivė $C_{trim}(u)$. Anga, atsiradusi iškarpytoje vietoje, nėra atvira, t.y. prijungtas paviršius $S_{n+1}(u, v)$ išilgai v parametro, kurio pradžia (žr.: Pav. 2.8.2):

$$S_{n+1}(0, v) = C_{trim}(u). \quad (\text{Išraiška 2.8.4})$$



Pav. 2.8.2: Iškarpančioji kreivė, nurodanti prijungto paviršiaus pradžia

Jei prijungtų paviršių, kurių tūrį reikia suskaičiuoti, yra m , tada nagrinėjami $S_n(u, v)$, $S_{n+1}(u, v)$... $S_{n+m}(u, v)$ paviršiai kaip ir bendru atveju.

Idealiu atveju kas žingsnį t apskaičiuojamos paviršiaus taškų koordinatės vienai v kreivei:

$$S(jt, it) = P_{ji}. \quad (\text{Išraiška 2.8.5})$$

Čia $i = 0, 1 \dots 1/t$, o $j = const$. Taško P_{ji} koordinatės skaičiuojamos (žr.: Išraiška 2.5.10), atsižvelgiant į *NURBS* kontrolinių taškų pozicijas, svorius, mazgų vektorius ir bazinių funkcijų laipsnį. Tiek 35, tiek *Išimtis 2* atveju į kreivę įsiterpia atkarpos, priklausančios pagalbinėms plokštumoms. *Išimtis 3* atveju į iškarpančias kreives neatsižvelgiama.

Apskaičiuotų taškų P_{ji} seka sudaro kreivės $S(jt, v)$ diskretų daugiakampį ($j = const$), kurio plotas yra $S_{S(jt, v)}$. Tada paviršiaus tūris tarp dviejų gretimų v kreivių apytiksliai lygus:

$$V_j = \frac{S_{S(jt, v)} + S_{S((j-1)t, v)}}{2} h_j, \quad (\text{Išraiška 2.8.6})$$

$$h_j = t \sum_{i=1}^{1/t} h_{ji}. \quad (\text{Išraiška 2.8.7})$$

Čia h_j yra vidutinis atstumas u kryptimi tarp kreivių $S(jt, v)$ ir $S((j-1)t, v)$.

Bendru atveju iš taškų P_{ji} sudarytas daugiakampis nebus plokščias. Vadinasi reikia rasti plokštumą į kurią būtų projektuojami daugiakampio taškai. Tokios plokštumos normalės vektorius lygus dviejų tos plokštumos vektorių sandaugai $\vec{n}_j = \vec{k}_j \times \vec{l}_j$. Vektoriai \vec{k}_j ir \vec{l}_j randami taip:

1. Seka P_{ji} padalinama į 3 lygias dalis:

a. I dalis $it \in [0; 1/3)$.

b. II dalis $it \in [1/3; 2/3)$.

c. III dalis $it \in [2/3; 1)$.

2. Randami visų dalių vidurio taškai P_{jI}, P_{jII} ir P_{jIII} pagal koordinatinių vidurkį.

3. Randami $\vec{k}_j = P_{jI} - P_{jIII}, \vec{l}_j = P_{jII} - P_{jIII}$. Abu vektoriai normalizuojami.

Visi P_{ji} taškai į plokštumą $n_{jx}x + n_{jy}y + n_{jz}z = 0$, kur (n_{jx}, n_{jy}, n_{jz}) yra normalizuoto vektoriaus \vec{n}_j koordinatės [2]:

$$P'_{ji} = P_{ji} \cdot \begin{bmatrix} n_{jy}^2 + n_{jz}^2 & -n_{jx}n_{jy} & -n_{jx}n_{jz} \\ -n_{jx}n_{jy} & n_{jx}^2 + n_{jz}^2 & -n_{jy}n_{jz} \\ -n_{jx}n_{jz} & -n_{jy}n_{jz} & n_{jx}^2 + n_{jy}^2 \end{bmatrix}. \quad (\text{Išraiška 2.8.8})$$

Taškas P'_{ji} yra modelio erdvės taškas suprojektuotas į plokštumą $n_{jx}x + n_{jy}y + n_{jz}z = 0$. Perskaičiuojamas vektorius $\vec{l}_j = \vec{n}_j \times \vec{k}_j$ vienas kitam statmeni vektoriai duotoje plokštumoje. Normalizavus \vec{k}_j ir \vec{l}_j vektorius, P'_{ji} taškus galima išreikšti projekcija į pastaruosius vektorius:

$$P''_{ji} = [P'_{ji} \cdot \vec{k}_j, P'_{ji} \cdot \vec{l}_j]. \quad (\text{Išraiška 2.8.9})$$

$P''_{ji} = [x''_{ji}, y''_{ji}]$ yra tašką P'_{ji} atitinkančios koordinatės $n_{jx}x + n_{jy}y + n_{jz}z = 0$ plokštumos erdvėje. Plokščio daugiakampio, sudaryto iš P'_{ji} taškų, plotas bus lygus [8]:

$$S_j = S_{S(jI, v)} \approx \frac{1}{2} \left| (y''_{j(1/t)} - y''_{j0})(x''_{j(1/t)} + x''_{j0}) + \sum_{i=1}^{1/t} (y''_{j(i-1)} - y''_{ji})(x''_{j(i-1)} + x''_{ji}) \right|. \quad (\text{Išraiška 2.8.10})$$

Procesą pakartojus $1/t$ kartų, gaunami visi v kreivių plotai S_j išilgai parametro $u, j = 0, 1 \dots 1/t$. Tūris tarp dviejų gretimų v kreivių skaičiuojamas pagal formulę (žr.: Išraiška 2.8.6). Viso paviršiaus ribojamas tūris yra:

$$V_k = \sum_{j=1}^{1/t} V_j. \quad (\text{Išraiška 2.8.11})$$

Diskretizavimo žingsnio t dydis priklauso nuo d reikšmės. Tegul maksimali t reikšmė lygi $1/9$ (vardiklis dalus iš 3). Jei A, B ir C yra paviršių rinkinį $S_1(u, v), S_2(u, v) \dots S_n(u, v)$ ribojančio stačiakampio gretasienio matmenys modelio erdvėje, išrikiuoti didėjimo tvarka, tada t reikšmė skaičiuojama pagal formulę:

$$t = \frac{d}{A\pi}. \quad (\text{Išraiška 2.8.12})$$

Algoritmas:

1. Skaičiuoti pradines reikšmes: A, t (Išraiška 2.8.12), $k = 1, i = 0, j = 0, V = 0$.
2. Kiekvienam paviršiui $S_k(u, v)$ skaičiuoti :
 - 2.1. Nustatyti $V_k = 0$.
 - 2.2. Su visais $jt \in [0; 1]$:
 - 2.2.1. Nustatyti $S_j = 0$.
 - 2.2.2. Su visais $it \in [0; 1]$:
 - 2.2.2.1. Skaičiuoti paviršiaus $S_k(u, v)$ taškus $P_{ji} = S_k(jt, it)$ (Išraiška 2.5.10).
 - 2.2.2.2. Jei P_{ji} yra už pagalbinės plokštumos ribos, tada P_{ji} prilyginti ribinei reikšmei.
 - 2.2.2.3. Jei $j > 0$, tada skaičiuoti h_j (Išraiška 2.8.7).
 - 2.2.3. Skaičiuoti vektorius $\vec{k}_j = P_{ji} - P_{jii}, \vec{l}_j = P_{jii} - P_{jiii}, \vec{n}_j = \vec{k}_j \times \vec{l}_j$ ir normalizuoti.
 - 2.2.4. Perskaičiuoti $\vec{l}_j = \vec{n}_j \times \vec{k}_j$ ir normalizuoti.
 - 2.2.5. Su visais $it \in [0; 1]$:
 - 2.2.5.1. Skaičiuoti taškų projekcijas P_{ji}' (Išraiška 2.8.8).
 - 2.2.5.2. Skaičiuoti taškų projekcijas $P_{ji}'' = [x_{ji}'', y_{ji}'']$ (Išraiška 2.8.9).
 - 2.2.5.3. Skaičiuoti S_j (Išraiška 2.8.10).
 - 2.2.6. Jei $j > 0$, tada skaičiuoti V_j (Išraiška 2.8.6).
 - 2.2.7. Jei $j > 0$, tada skaičiuoti $V_k = V_k + V_j$ (Išraiška 2.8.11).
 - 2.3. Skaičiuoti $V = V + V_k$ (Išraiška 2.8.2).
3. Rezultatas: V .

2.9. Modelio koregavimo (nurodytoje srityje) algoritmas

Uždavinys: Keičiamos paviršių rinkinio $S_1(u, v), S_2(u, v) \dots S_n(u, v)$ srities A proporcijos pasirinktos koordinatinių ašies kryptimi, manipuluojant kontroliniais taškais $CP_A \in A$. Atsparos taško pozicija P_0 gali būti nurodoma. Taip pat gali būti nurodomi paviršiai $S_1(u, v) \dots S_m(u, v)$, kuriuos įtakos srities A proporcijų keitimas.

Analizė: Prieš koreguojant modelio sritį A būtina pasižymėti visus kontrolinius taškus, su kuriais bus atliekama operacija. Taškai gali priklausyti vienam ar keliems paviršiams, be to, sritis gali apimti tik dalį paviršiaus. Tarkim koreguotinai sričiai priklausantys kontroliniai taškai yra CP_A , kurių kiekis yra n_{CPA} . Tada, jei vartotojas nenurodo kitaip, atsparos taškas bus geometrinis visų srities taškų vidurkis:

$$P_0 = \frac{1}{n_{CPA}} \sum_{CP_A \in A} CP_A. \quad (\text{Išraiška 2.9.1})$$

Srities A proporcijų keitimas gali įtakoti srities B kontrolinių taškų pozicijas. B sritis reikalinga tam, kad išlaikyti nedeformuojamų modelio sričių pozicijas, deformuojamosios atžvilgiu. Šiom sritim galioja tokios savybės:

- $A \cap B = \emptyset$, kur \emptyset yra tuščia sritis.
- $A \cup B = C$, kur C yra sritis, kuriai taikomos transformacijos, koregavimo operacijos metu.
- $A \subseteq D, B \subseteq D, C \subseteq D$, kur D gaminio modelio sritis (visi iš bylos nuskaityti kontroliniai taškai).
- $CP_B \in B$.
- $\bar{A} = D - A$, \bar{A} sričiai priklauso visi modelio kontroliniai taškai, išskyrus CP_A .

Sritis A turi būti vientisa, t.y. sritį ribojančiame stačiakampyje gretasienyje negali būti sričių priklausančių \bar{A} . Sritį ribojantis stačiakampis gretasienis apibrėžiamas ribinių CP_A :

- Srities ribos išilgai Ox ašies: $x_{Amin} = \min(x_{CP_A i}), x_{Amax} = \max(x_{CP_A i})$.
- Srities ribos išilgai Oy ašies: $y_{Amin} = \min(y_{CP_A i}), y_{Amax} = \max(y_{CP_A i})$.
- Srities ribos išilgai Oz ašies: $z_{Amin} = \min(z_{CP_A i}), z_{Amax} = \max(z_{CP_A i})$.

Čia $(x_{CP_A i}, y_{CP_A i}, z_{CP_A i})$ yra taško $CP_A i$ koordinatės, kur $i = 1, 2 \dots n_{CPA}$.

Prieš operaciją srities A proporcijos (Ox, Oy, Oz) ašių atžvilgiu $(1, 1, 1)$. Po operacijos sritis deformuojama, naudojant mastelio (*scaling*) transformaciją (naudojamos homogeninės koordinatės) [4], pagal S_x, S_y, S_z reikšmes, kurias pasirenka vartotojas. Prieš transformaciją sritis A perkeliama,

naudojant poslinkio (*moving*) transformaciją, kad P_0 koordinatės (P_{0x} , P_{0y} , P_{0z}) sutaptų su koordinatių sistemos pradžia ($0, 0, 0$):

$$CP'_{Ai} = CP_{Ai} \cdot \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ -S_x P_{0x} & -S_y P_{0y} & -S_z P_{0z} & 1 \end{bmatrix}. \quad (\text{Išraiška 2.9.2})$$

Čia $i = 1, 2 \dots n_{CPA}$. Po transformacijos CP'_{Ai} yra paslinkti, kad sutapatinti P_0 su koordinatių sistemos pradžia. Todėl reikia dar vienos poslinkio transformacijos, kad gražinti sritį A į pradinę vietą. Padauginus transformacijos matricą (žr.: *Išraiška 2.9.2*) iš gražinimo atgal transformacijos matricos gaunamas rezultatas:

$$CP''_{Ai} = CP_{Ai} \cdot \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ (1-S_x)P_{0x} & (1-S_y)P_{0y} & (1-S_z)P_{0z} & 1 \end{bmatrix}. \quad (\text{Išraiška 2.9.3})$$

Srities B transformacija nėra bendra visai sričiai, todėl sritis skaidoma į smulkesnes sritis (poaibius):

- $CP_{Bi} \in B_{minx}$, kai $x_{CPBi} < x_{Amin}$.
- $CP_{Bi} \in B_{maxx}$, kai $x_{CPBi} > x_{Amax}$.
- $CP_{Bi} \in B_{miny}$, kai $y_{CPBi} < y_{Amin}$.
- $CP_{Bi} \in B_{maxy}$, kai $y_{CPBi} > y_{Amax}$.
- $CP_{Bi} \in B_{minz}$, kai $z_{CPBi} < z_{Amin}$.
- $CP_{Bi} \in B_{maxz}$, kai $z_{CPBi} > z_{Amax}$.

Kontrolinis taškas CP_{Bi} priklauso vienai, dviem arba trimis B poaibiams, kai $i = 1, 2 \dots n_{CPB}$. Tačiau tas pats kontrolinis taškas negali priklausyti dviem poaibiams su tos pačios ašies ženklu (pvz.: jei $CP_{Bi} \in B_{maxz}$, tai $CP_{Bi} \notin B_{minz}$).

Bendra poslinkio transformacijos matrica B sričiai atrodytų taip:

$$CP'_{B \text{ Poaibis } i} = CP_{B \text{ Poaibis } i} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix}. \quad (\text{Išraiška 2.9.4})$$

Konkrečios T_x , T_y ir T_z reikšmės pateiktos (žr.: *Lentelė 2.9.1*). Jei Taškas priklauso keliems poaibiams, tuomet jam taikomos kelios transformacijos kiekvienai ašiai arba viena bendra visom ašim.

B poaibis	T_x	T_y	T_z
$B_{min x}$	$(x_{min} - P_{0x})S_x$	0	0
$B_{max x}$	$(x_{max} - P_{0x})S_x$	0	0
$B_{min y}$	0	$(y_{min} - P_{0y})S_y$	0
$B_{max y}$	0	$(y_{max} - P_{0y})S_y$	0
$B_{min z}$	0	0	$(z_{min} - P_{0z})S_z$
$B_{max z}$	0	0	$(z_{max} - P_{0z})S_z$

Lentelė 2.9.1: B poaibių poslinkio transformacijos reikšmės

Algoritmas:

1. Pažymėti nurodytos srities A taškus kaip CP_A , skaičiuoti n_{CPA} .
2. Pažymėti nurodytos srities B taškus kaip CP_B , skaičiuoti n_{CPB} .
3. Rasti x_{Amin} , x_{Amax} , y_{Amin} , y_{Amax} , z_{Amin} , z_{Amax} reikšmes.
4. Skaidyti sritį B į poaibius $B_{min x}$, $B_{max x}$, $B_{min y}$, $B_{max y}$, $B_{min z}$, $B_{max z}$.
5. Fiksuoti vartotojo pasirinktas S_x , S_y , S_z reikšmes.
6. Jei vartotojas nenurodo P_0 , tada skaičiuoti P_0 (Išraiška 2.9.1).
7. Skaičiuoti T_x , T_y , T_z (Lentelė 2.4.1).
8. Su visais $i = 1, 2 \dots n_{CPA}$:
 - 8.1. Skaičiuoti CP_{Ai}'' (Išraiška 2.9.3).
9. Su visais $i = 1, 2 \dots n_{CPB}$:
 - 9.1. Skaičiuoti CP_B^i pagal tai, kokiam poaibiui ($CP_{B\text{Poaibis } i}^i$) priklauso taškas (Išraiška 2.9.4).
10. Rezultatas: naujos kontrolinių taškų pozicijos $CP_A = CP_A''$ ir $CP_B = CP_B^i$.

2.10. Pakavimo dvimatėje erdvėje algoritmas

Uždavinys: Duota aibė kontūrų $f \in F$ ir stačiakampė plokštuma A , kurią reikia užpildyti kontūrais iš aibės F , minimizuojant ploto sąnaudas. Taip pat duotas minimalus atstumas tarp kontūrų d . Pakuojama, keičiant tarpusavio kontūrų orientaciją ir mažinant atraižų plotą.

Analizė: Kiekviena figūra $f \in F$ turi bent tris viršūnes ir bent vieną kontūrą. Kontūrą sudaro viršūnės, sujungtos briaunomis. Kontūras turi būti uždaras, nebūtinai iškilusis, o jo briaunos gali kirstis tik viršūnių taškuose. Jei f turi daugiau nei vieną kontūrą, tai visi kontūrai, išskyrus pirmąjį, yra vidiniai, apibrėžiantys figūros f vidaus ertmes.

Duotas lapas A (stačiakampio formos darbinis laukas), kuris apibrėžtas dviejų taškų – kairiojo apatinio ir dešiniojo viršutinio bei įrankio diametras d .

Reikia išdėlioti $f_i \in F$ lape A taip, kad tarp f_i briaunų ir f_j briaunų atstumas būtų ne mažesnis kaip d , kai $i = 1, 2.. N, j = 1, 2.. N, i \neq j$. Čia N – figūrų kiekis aibėje F . Figūras f_i , dedant į lapą A , galima orientuoti (sukti) bet koku kampu.

Užduotis sprendžiama, pradedant nuo figūrų sekos S sudarymo pagal kriterijų l . Kriterijus l yra atstumas tarp labiausiai nutolusių figūros taškų. Rikiuojama l mažėjimo tvarka. Į seką S neįtrauktos figūros, kurių $l^2 > \Delta x_A^2 + \Delta y_A^2$ (lapo A įstrižainė, kai kairiojo apatinio A taško koordinatės $(0, 0)$).

Vėliau seka S skaidoma į smulkesnes sekas S_i , kurių dydis yra r ; apjungtos smulkesnės sekos, atsižvelgiant į eiliškumą, sudaro seką S . Figūros sekose S_i eiliškumas keičiamas atsitiktine tvarka, tačiau smulkiųjų sekų tarpusavio eiliškumas nepakinta. Atsižvelgiant į eiliškumą, sudaroma seka S_r , $|S_r| = |S|$. Atsitiktinumo faktorius leidžia rasti geriausią sprendinį per duotą laiką.

Atlikus šiuos pradinius veiksmus, pirmoji figūros iš sekos S_r pozicionuojama ir orientuojama lape A ($\Delta x_A \geq \Delta y_A$). Surandamas iškilusis kampas α artimiausias $\pi/2$ ir figūra pasukama taip, kad α sutaptų su lapo A $(0+d/2, 0+d/2)$ tašku (apatinis kairysis kampas). Jei iškilių kampų nėra – tuomet ieškoma neiškilaus kampo ir jis orientuojamas į apatinį kairįjį lapo A kampą bei pozicionuojama taip, kad figūros taškai su $\min(x) = d/2, \min(y) = d/2$.

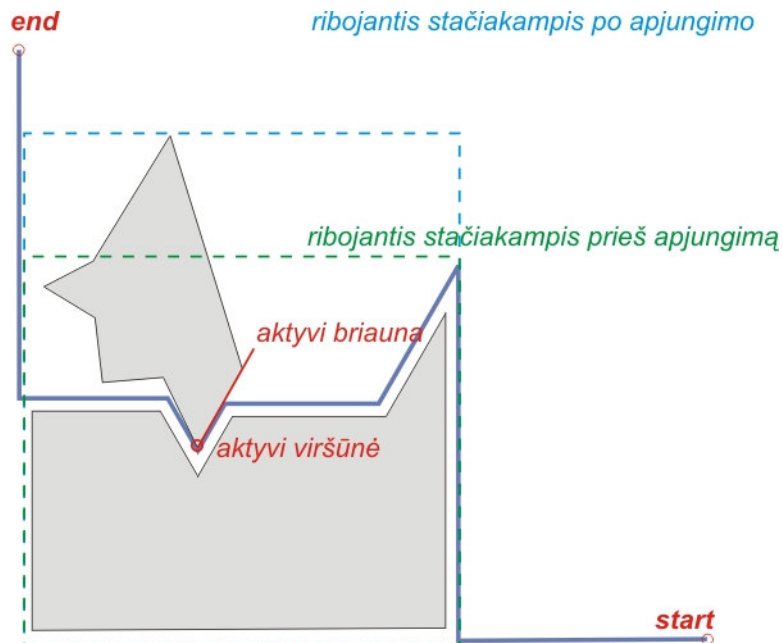
Laikoma, kad ribojantis stačiakampis apibrėžia lapo A sritį, kuri bus sunaudota, apdirbant detales (išpjaunant figūras iš lapo). Ribojantis stačiakampis dalina A į dvi dalis: vidinę – su figūromis ir atraižomis, išorinę – nepanaudoto ploto.

Suradus figūros poziciją ir orientaciją sudaroma dengiančiojo kontūro architektūra, atsižvelgiant į d :

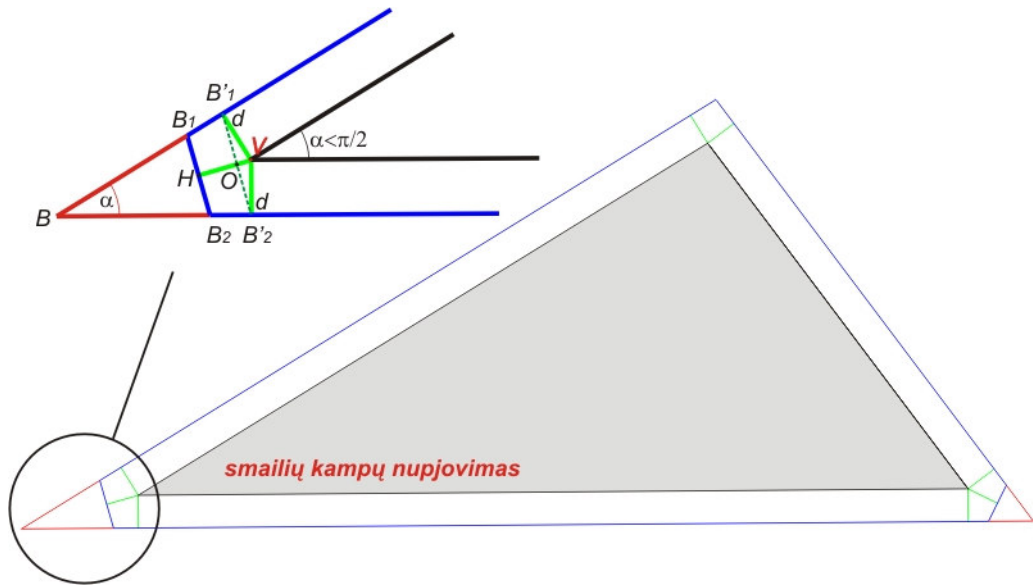
- Jei kontūras dar neegzistuoja, tuomet jis kuriamas apeinant pirmąją figūrą prieš laikrodžio rodyklę (vidinį kontūrą – pagal laikrodžio rodyklę) ir brėžiant iš dešinės per d atidėta kontūro tiesę lygiagrečiai figūros kraštinei. Tiesės, susikertančios kampu $\geq \pi/2$, sudaro išorinio kontūro atkarpas. Jei tiesės susikerta mažesniu nei $\pi/2$ kampu (išoriniam kontūrai) – smailūs kampai nupjaunamas, atstumu d nuo figūros kampo (žr.: Pav. 2.10.2).
- Jei kontūras egzistuoja – brėžiamas laikinas antrasis kontūras laikantis ankščiau paminėtų principų. Nustatomi išorinio ir laikino kontūrų kirtimosi taškai ir vykdoma jungimo operacija (žr.: Pav. 2.10.3).
- Minimizuojamas kontūro viršūnių skaičius – pašalinami kampai artimi π .

Architektūra – masyvas su dengiančio kontūro kampų reikšmėmis, briaunų prieš laikrodžio rodyklę kryptimi ilgais ir vėliavėlėmis. Briaunos lietimasis ar kirtimasis su lapo A riba sudaro dar vieną kampą su briauna, kuri lygiagreti kertamai/liečiamai A kraštinei, jos galas yra nesujungtas su likusiu kontūru bei turi *start/end* vėliavėlę (žr.: Pav. 2.10.1). *start* ir *end* viršūnės susijungia tik tada, kai pilnai užpildytas A (ignoruojant vidinius kontūrus). Šie taškai neįeina į nagrinėjamas viršūnes, nes nesudaro kampo. Vidiniai kontūrai prasideda iš baigiasi tuo pačiu tašku.

Apie gauto rezultato kokybę bus sprendžiama iš ribojančio stačiakampio ploto dydžio (žr.: Pav. 2.10.1).

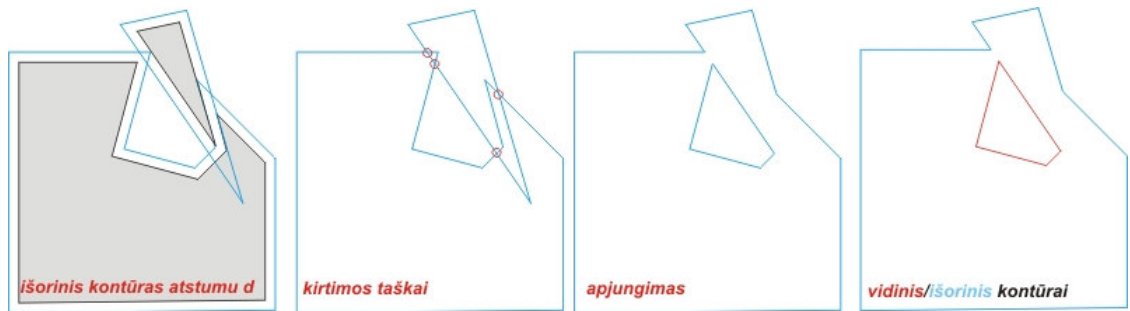


Pav. 2.10.1: Start/end viršūnės, ribojantis stačiakampis



Pav. 2.10.2: Smailių kampų nupjovimas

Smailūs kampai užima kelis kartus daugiau ploto, nei jo reiktų įrankiui su diametru d , todėl išorinio dengiančiojo kontūro smailūs kampai keičiami dviem bukais, kaip parodyta (Pav. 7.2): kampas B pakeičiamas kampais B_1 ir B_2 . B_1 ir B_2 taškai gaunami atidėjus lygiagretes atstumu d nuo duotų figūros briaunų. $B_1V = B_2V = HV = d$, $HO = (1 - \sin(\alpha/2))d$. Belieka atidėti lygiagrete HO atstumu nuo atkarpos B_1B_2 ir rasti kirtimosi taškus B_1 ir B_2 .



Pav. 2.10.3 dengiančiojo kontūro konstravimas

Dviejų kontūrų apjungimo operacija gali sugeneruoti papildomų vidinių kontūrų. Apjungimo algoritmas:

1. Rasti kontūrų C_1 (bendro) ir C_2 (laikino) kirtimosi taškus P_k ir įsiminti kurios kontūrų briaunos kirtosi: i, j , kur $k = 1, 2 \dots N_{kirt}$, $i = 1, 2 \dots N_{C1}$, $j = 1, 2 \dots N_{C2}$
2. Išrikiuoti $[P_k, i, j]$ indekso i didėjimo tvarka, jei indeksai sutampa, tuomet pirmesnis yra taškas, arčiau aktyvios viršūnės.
3. Išorinis kontūras:
 $start \xrightarrow{(+i)} [P_1, i, j], P_1, [P_1, i, j]+1 \xrightarrow{(+j)} [P_{N_{kirt}}, i, j], P_{N_{kirt}}, [P_{N_{kirt}}, i, j]+1 \xrightarrow{(+i)} end$
4. Panaikinami $P_1, P_{N_{kirt}}$.
5. Vidiniai kontūrai:

- a. $P_1, [P_1, i, j]+1 \xrightarrow{(+i)} [P_2, i, j], P_2, [P_2, i, j] \xrightarrow{(+j)} [P_1, i, j], P_1$.
- b. Panaikinami P_1, P_2 .

Aprašyti algoritmai nenagrinėja figūrų su lankais, tačiau lankus pavertus atkarpomis pagal užduotą žingsnį, algoritmai tinkami.

Algoritmas:

1. Figūrų iš aibės F sekos S sudarymas pagal kriterijų l , sudaryti seka S_r pagal parametą r .
2. Maišyti figūrų eilės tvarką S_r sekoje.
3. Su visais $i = 1, 2 \dots N$:
 - 3.1. Pozicionuoti ir orientuoti f_i iš S_r lape A ($\Delta x_A \geq \Delta y_A$)
 - 3.2. Jei $i = 1$, tada tikrinama ar figūra telpa į lapą A :
 - 1.1.1. Netelpa išilgai Δy_A – pasukti objektą $-\pi/2$, pozicionuoti taip kad figūros taškai $\min(x) = d/2, \min(y) = d/2$.
 - 1.1.2. Netelpa išilgai Δx_A – orientuoti objekto l lygiagrečiai ($\Delta x_A, \Delta y_A$) vektoriui, pozicionuoti $\min(x) = d/2, \min(y) = d/2$.
 - 1.1.3. Netelpa išilgai ($\Delta x_A, \Delta y_A$) – pašalinti iš sekos S_r , sumažinti N per 1, grįžti į 3.1 žingsnį.
 - 3.3. Jei $i > 1$, Tikrinti ar figūra telpa į vidinį kontūrą, jei ne – į išorinį:
 - 3.3.1. Tikrinami visi figūros kampai visų bendro kontūro kampų atžvilgiu kiekvieną kartą orientuojant taip, kad „aktyvi“ figūros kraštinė (prieš laikrodžio rodyklę nuo tikrinamos viršūnės) būtų lygiagreti kontūro aktyviai kraštinei (pagal laikrodžio rodyklę nuo tikrinamo kontūro kampo).
 - 3.3.2. Vidinio kontūro atžvilgiu: jei nekerta kontūro, pasirenkamas tas variantas, kuris palieka mažiausią „skylę“.
 - 3.3.3. Išorinio kontūro atžvilgiu: jei figūra neišeina iš už A ribų ir nekerta bendro kontūro, pasirenkamas variantas, kurio mažiausias *ribojančio stačiakampio* (stačiakampis apibrėžiantis kontūrą, kurio briaunose yra kontūro viršūnių $\min(x), \min(y), \max(x), \max(y), start/end$ viršūnės neįeina (žr.: Pav. 2.10.1) figūros su bendru kontūru ploto prieauglis.
 - 3.4. Atnaujinti dengiančio kontūro architektūrą.
4. Rezultatas: figūros iš sekos S_r išdėliotos lape A .

2.11. Dvimačio kontūro koregavimo algoritmai

Šiame skyriuje nagrinėjami du skirtingos paskirties kontūro koregavimo algoritmai: kontrolinių taškų kiekio minimizavimo ir splaininės kreivės vertimas atkarpomis bei lankais.

2.11.1. Kontrolinių taškų kiekio minimizavimas

Užduotis: Duota *NURBS* kreivė $C(u)$ su kontroliniais taškais $P_i, i = 1, 2 \dots n$. Čia n yra kontrolinių taškų kiekis. Taip pat duotas mazgų vektorius U , svoriai w_i ir bazinės funkcijos laipsnis p . Vartotojas pasirenka galimos atstumo paklaidos ε , galimos kampo paklaidos γ ir mažiausio atstumo tarp kontrolinių taškų d reikšmės ($d \geq \varepsilon$). Reikia sudaryti $C'(u)$ su kontroliniais taškais $P'_i, i = 1, 2 \dots n'$, kur $C'(u) \approx C(u)$ ir $n' \leq n$.

Analizė: Kreivės, sugeneruotos įvairių *CAD* programų, geometrinės konstrukcijos keitimo ar duomenų eksportavimo metu paprastai turi perteklinių kontrolinių taškų. Pertekliniai kontroliniai taškai yra tokie taškai, kurių naikinimas nekeičia arba nežymiai keičia kreivę $C(u)$.

Dėl bazinės funkcijos *NURBS* kreivė būna glotni ir neturi aštrių kampų. Norint modeliuoti aštrų kampa būtina sukurti kelis kontrolinius taškus toje pačioje pozicijoje. Tokiu kontrolinių taškų skaičius lygus p . Jei $p = 1$, tuomet kreivė tampa laužte. Paprastai $p = 3$. Tokie taškai, dėl galimų paklaidų, nebūtinai yra toje pačioje pozicijoje, tačiau atstumas tarp jų gali būti labai mažas. Šis atstumas privalo būti mažesnis už ε .

Dėl galimo aštraus kampo formavimo fakto, negalima naikinti visų kolinearių (artimų kolineariams) taškų, jei tarp jų atstumas mažesnis už ε . Kampo formavimo faktas tikrinamas taip:

- Jei p kaimyninių kontrolinių taškų $P_i, P_{i+1} \dots P_{i+p-1}, i = 2, 3 \dots n - p$, nuo gretimo taško nutolę atstumu mažesniu už ε , tada matuojamas kampas α tarp atkarpų $P_{i-a}P_{i+r}$ ir $P_{i+r}P_{i+p+b}$. Čia $r = p/2$, jei p yra lyginis ir $r = (p-1)/2$ kitu atveju. Be to, galioja apribojimai $1 \leq i - a < i, i < i + b \leq n, |P_{i-a}P_{i+r}| > d$ ir $|P_{i+r}P_{i+p+b}| > d$.
- Jei $|\pi - \alpha| > \gamma$, tada visų kaimyninių $P_i, P_{i+1} \dots P_{i+p-1}$ pozicijos prilyginamos taško P_{i+r} pozicijai ir panaikinami taškai tarp P_{i-a} ir P_{i+p+b} . Kitu atveju panaikinami visi kaimyniniai taškai tarp P_{i-a} ir P_{i+r} bei P_{i+r} ir P_{i+p+b} .

Neuždaros kreivės atveju pirmasis ir paskutinis kontroliniai taškai neformuoja kampų. Jei kreivė uždara, tada visi taškai gali formuoti kampus, ir ankščiau paminėtas tikrinimo būdas tinka, pakoregavus indeksavimo sistemą.

Visiems taškams, kurie neformuoja kampų, tikrinamas kolinearumo faktas:

- Matuojamas α kampas tarp dviejų atkarpų, kurias sudaro trys kaimyniniai kontroliniai taškai $P_{i-a}, P_i, P_{i+b}, i = 2, 3 \dots n - 1$, kai $1 \leq i - a < i, i < i + b \leq n, |P_{i-a}P_{i+r}| > d$ ir $|P_{i+r}P_{i+p+b}| > d$.
- Jei $|\pi - \alpha| \leq \gamma$, tada pašalinamas taškas P_i .

Atstumo tarp taškų P_1 ir P_2 su koordinatėmis (x_{P1}, y_{P1}, z_{P1}) ir (x_{P2}, y_{P2}, z_{P2}) skaičiavimo formulė:

$$d_{P1P2} = \sqrt{(x_{P1} - x_{P2})^2 + (y_{P1} - y_{P2})^2 + (z_{P1} - z_{P2})^2}. \quad (\text{Išraiška 2.11.1})$$

Kampo tarp atkarpų, kurių galų koordinatės $(x_{P1}, y_{P1}, z_{P1}), (x_{P0}, y_{P0}, z_{P0})$ ir (x_{P2}, y_{P2}, z_{P2}) , kur P_0 yra susikirtimo taškas, skaičiavimo formulė:

$$\alpha = \arccos\left(\frac{(x_{P1} - x_{P0})(x_{P2} - x_{P0}) + (y_{P1} - y_{P0})(y_{P2} - y_{P0}) + (z_{P1} - z_{P0})(z_{P2} - z_{P0})}{d_{P0P1}d_{P0P2}}\right). \quad (\text{Išraiška 2.11.2})$$

Čia d_{P0P1} – atstumas tarp (x_{P1}, y_{P1}, z_{P1}) ir (x_{P0}, y_{P0}, z_{P0}) , o d_{P0P2} – tarp (x_{P0}, y_{P0}, z_{P0}) ir (x_{P2}, y_{P2}, z_{P2}) taškų.

Algoritmas:

1. Nustatyti $P_i, U, w_i, n, p, \gamma, \varepsilon, d$ reikšmes, priskirti $t = 0$.
2. Sukurti n ilgio masyvą C_{rn} , kurio visi nariai lygūs $i = 1, 2 \dots n$ ir n ilgio masyvą D kurio nariai lygūs 0 . Priskirti $i = 1$.
3. Kol $i \leq n$:
 - 3.1. Matuoti atstumą $d_{P_{i+1}P_{i+t+1}}$ tarp P_{i+1} ir P_{i+t+1} (Išraiška 2.11.1).
 - 3.2. Jei $d_{P_{i+1}P_{i+t+1}} < \varepsilon$ ir $t < p - 1$ ir $i + t < n$, tada priskirti $t = t + 1$ ir grįžti į 3.1. žingsnį.
 - 3.3. Jei $t > 0$, tada:
 - 3.3.1. Jei t yra lyginis, tada priskirti $r = t / 2$.
 - 3.3.2. Jei t yra nelyginis, tada priskirti $r = (t - 1) / 2$.
 - 3.3.3. Visų taškų P_j koordinatės prilyginti P_{i+r} taško koordinatėms, kai $j = i + 1, i + 2 \dots i + t$.
 - 3.3.4. Visus C_{rn_j} narius prilyginti $i + r$ reikšmei, kai $j = i + 1, i + 2 \dots i + t$.
 - 3.4. Priskirti $D_{i+r} = d_{P_{i+1}P_{i+t+1}}$.
 - 3.5. Priskirti $i = i + t + 1$.
 - 3.6. Priskirti $t = 0$.
4. Priskirti $i = 1$.
5. Kol $i \leq n - 2$:
 - 5.1. Rasti mažiausią t , su kuriuo $D_{i+t+1} \neq 0$.
 - 5.2. Priskirti $A = P_i, B = P_j$ ir $C = P_k$, kur $j = C_{rn_{i+t+1}}, D_k \neq 0, k \geq i + t + 2, k \leq n$.

- 5.3. Matuoti kampą α tarp atkarpų AB ir BC (Išraiška 2.11.2).
- 5.4. Jei $D_i + D_{i+t+1} < d$ arba $|\pi - \alpha| < \gamma$, tada:
- 5.4.1. Su visais j , su kuriais $Crn_j = i+t+1$: priskirti visus $Crn_j = 0$.
- 5.4.2. $i = i+t+k+2$.
- 5.5. Kitu atveju:
- 5.5.1. $i = i + t + 1$.
6. Jei $m < p + 1$, kur m yra kiekis $Crn_j \neq 0, j = 1, 2 \dots n$, tada paskutiniai $(p - m - 1)$ $Crn_j = 0$ nariai prilyginami $Crn_j = j$.
7. Su visais i , kur $Crn_i = 0$:
- 7.1. Pašalinti visus P_i, w_i .
- 7.2. Jei $U_i = 0$, tai $U_{i+1} = 0$.
- 7.3. Jei $U_i = 1$, tai $U_{i-1} = 0$.
- 7.4. Pašalinti U_i .
8. Rezultatas: minimizuota kontrolinių taškų seka P_i su mazgų vektorium U ir svoriais w_i .

2.11.2. **Splaininės kreivės vertimas į atkarpas ir lankus**

Užduotis: Duoti kontroliniai taškai P_i , svoriai w_i , bazinės funkcijos laipsnis p ir mazgų vektorius U , aprašantis kreivę $C(u)$. Vartotojas pasirenka diskretizavimo žingsnį d , galimos kampo paklaidos γ ir vertimo tipą (atkarpas arba atkarpas su lankais). Reikia kreivę paversti atkarpomis (ir lankais).

Analizė: Apskaičiuojamas diskretizavimo žingsnis parametrinėje erdvėje t (žr.: Išraiška 2.8.12). Jei kreivė verčiama tik į atkarpas, tada apskaičiuojami tikslūs kreivės taškai $C(it)$, kur $i = 1, 2 \dots 1/t$ (žr.: Išraiška 2.5.8), ir traktuojami kaip atkarpų galų taškai.

Jei kreivė verčiama į atkarpas bei lankus, tada apskaičiuojami trys kaimyniniai taškai $C(it)$ ir matuojamas kampas α (žr.: Išraiška 2.11.2), kuri sudarytų atkarpos nubrėžtos per pirmąjį ir antrąjį bei trečiąjį ir antrąjį taškus. Jei $|\pi - \alpha| \leq \gamma$, tada generuojama viena atkarpa, jei $\alpha > \pi/2$, tada generuojamos dvi atkarpos (nes gaunamas $>180^\circ$ lankas). Atkarpos generuojamos ir tuo atveju, jei vienos iš atkarpų, nubrėžtų per pirmąjį ir antrąjį bei trečiąjį ir antrąjį taškus, ilgis daugiau nei tris kart didesnis už kitos. Kitu atveju generuojamas lankas, einantis per tuos tris taškus.

Algoritmas:

1. Nustatyti $P_i, U, w_i, p, \gamma, d$ reikšmes, sukurti masyvą Pnt , kur bus saugomos apskaičiuotos taškų pozicijos ir masyvą Ent , kur bus saugomas generuojamo objekto tipas.
2. Apskaičiuoti t (Išraiška 2.8.12).
3. Su visais $i = 3, 4 \dots 1/t$:

- 3.1. Jei taškas A nesuskaičiuotas, tada skaičiuoti $A = C((i-2)t)$ ir $B = C((i-1)t)$ (Išraiška 2.5.8).
- 3.2. Skaičiuoti tašką $C = C(it)$ (Išraiška 2.5.8).
- 3.3. Matuoti kampą α (Išraiška 2.11.2).
- 3.4. Jei verčiame tik atkarpomis, tada:
 - 3.4.1. Jei $|\pi - \alpha| > \gamma$, tada Pnt masyve išsaugoti B reikšmę bei Ent masyve pažymėti atkarpos tipą. Priskirti $A = B$ bei $B = C$.
 - 3.4.2. Kitu atveju, priskirti $B = C$.
- 3.5. Kitu atveju:
 - 3.5.1. Jei $\alpha < \pi/2$ arba $d_{AB}/d_{BC} < 0,33$ ir $d_{AB}/d_{BC} > 3$, tai Pnt masyve išsaugoti B reikšmę bei Ent masyve pažymėti atkarpos tipą. Priskirti $A = B$ ir $B = C$ (Išraiška 2.11.1).
 - 3.5.2. Jei $\alpha \geq \pi/2$ ir $\alpha \leq \pi - \gamma$, tai Pnt masyve išsaugoti B ir C reikšmes bei Ent masyve pažymėti lanko tipą. Priskirti $A = C$. Priskirti $i = i + 1$. Apskaičiuoti $B = C(it)$.
4. Rezultatas: masyvuose Pnt ir Ent saugoma informacija apie generuojamus objektus vietoje $C(u)$.

3. GAMINIO MODELIO APDOROJIMO PROJEKTAVIMAS

3.1. Reikalavimų sistemai specifikacija

Modelis nuskaitymas iš *IGES* bylos ir po redagavimo vėl išsaugojamas *IGES* byloje. Programa neturi specifinio savo failų formato, o duomenys redagavimo metu saugojami operatyviojoje atmintyje. Galimas geometrinių parametrų išsaugojimas tekstiniame faile.

Dėl darbo su grafika specifikos keliami reikalavimai aparatūriniai įrangai:

- Vaizdo plokštė: aukšti, 3D spartinimas, bent 128MB vaizdo operatyviosios atminties (*ATI Radeon*).
- Procesorius: vidutiniai, apie 2 GHz (*AMD, Intel*).
- Operatyvioji atmintis: vidutiniai – aukšti, virš 512 MB RAM.
- Operacinė sistema: *Windows XP*.

Reikalavimai duomenims:

- Programinė įranga skaito ir rašo duomenis į *IGES* formato bylas.
- Palaikomos geometrinės esybės: lankas, atkarpa, *NURBS* kreivė, *NURBS* paviršius, iškarpytasis *NURBS* paviršius.
- Duomenų mainai testuoti su *MasterCAM* ir *3D Studio Max* sistemomis, kitų sistemų sugeneruotos *IGES* bylos gali būti interpretuojamos nekorektiškai.

3.2. Duomenų struktūra

Sistemos duomenims išsaugoti naudojama duomenų bazė „*Absolute Database for C++ Builder*“ (toliau *AB*). *AB* duomenų bazė tai *C++ Builder* komponentas, kuris leidžia įdiegti duomenų bazę į programinę įrangą. Šios duomenų bazės privalumai:

- *AB* duomenų bazė yra integruojama į pačią programinę įrangą, todėl jos nereikia papildomai įdiegti į kliento kompiuterį.
- Labai greitas duomenų apdorojimas.
- Duomenų bazes galima naudoti kaip duomenų bylas.
- Paprasta stebėti duomenis apdorojant gaminio modelį.

Sistemos duomenys skirstomi į 2 dalis:

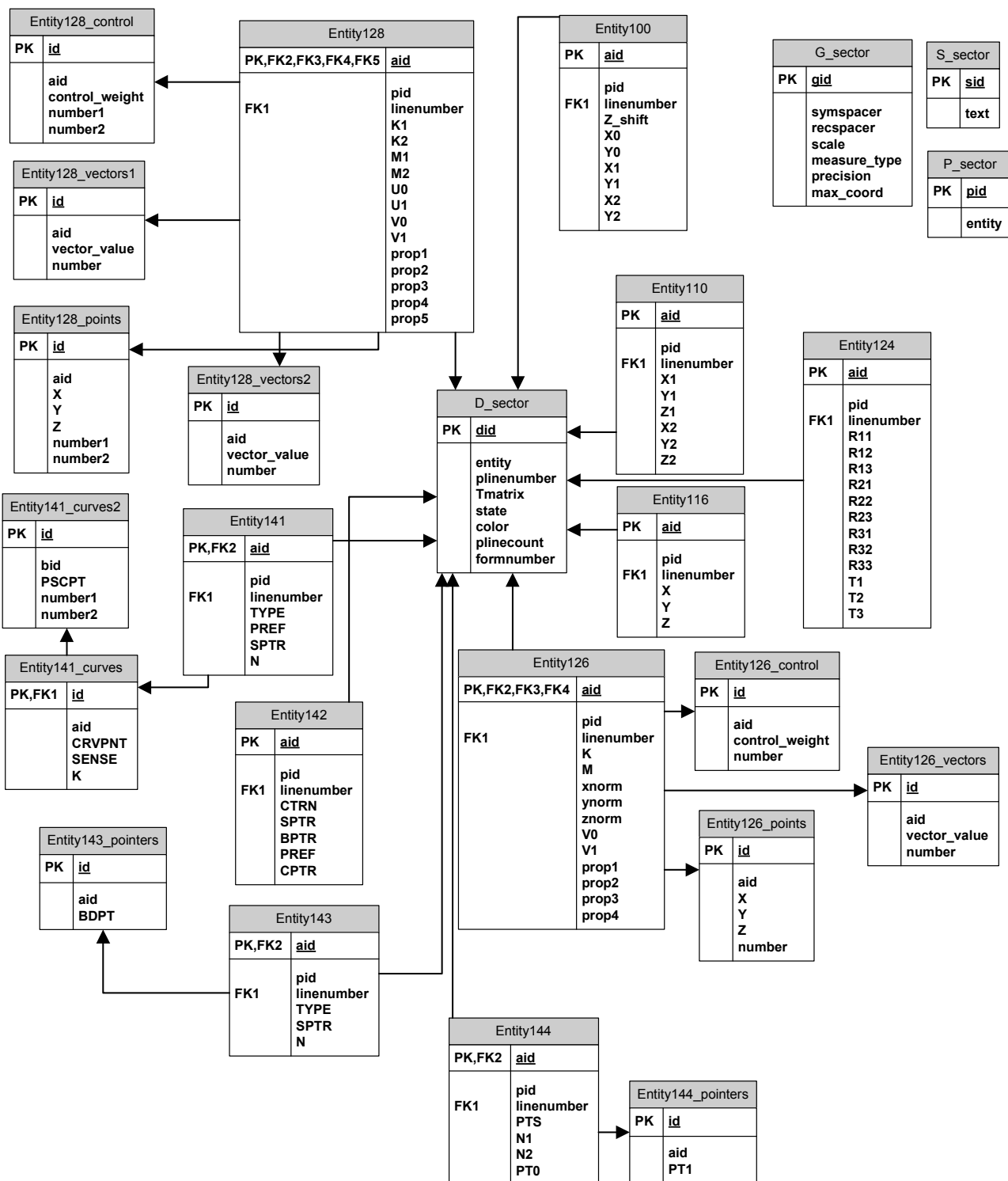
1. Gaminio modelio duomenys.
2. Pagalbiniai duomenys.

Gaminio modelio duomenys yra nuskaitomi iš *IGES* duomenų bylos (*IGES* formatas aprašytas 2.6 skyriuje). Pagalbiniai duomenys yra nustatomi programinės įrangos viduje.

Gaminio modelio duomenys išsaugojami duomenų bazėje pagal paruoštą šabloną. Kiekvienai nuskaitytai *IGES* bylai yra kuriama nauja duomenų bazė (pagal gaminio modelio duomenų bazės šabloną). Ši bazė yra pervardinama į nuskaitytos *IGES* bylos pavadinimą. Pagalbiniams duomenims naudojama atskira duomenų bazė.

3.2.1. Gaminio modelio duomenys

Gaminio modelio duomenų bazės šablono schema pavaizduota *Pav. 3.2.1.*



Pav. 3.2.1: Gaminio modelio duomenų bazės šablono schema

Gaminio modelio duomenų bazės esybės pateiktos (žr.: Lentelė 3.2.1).

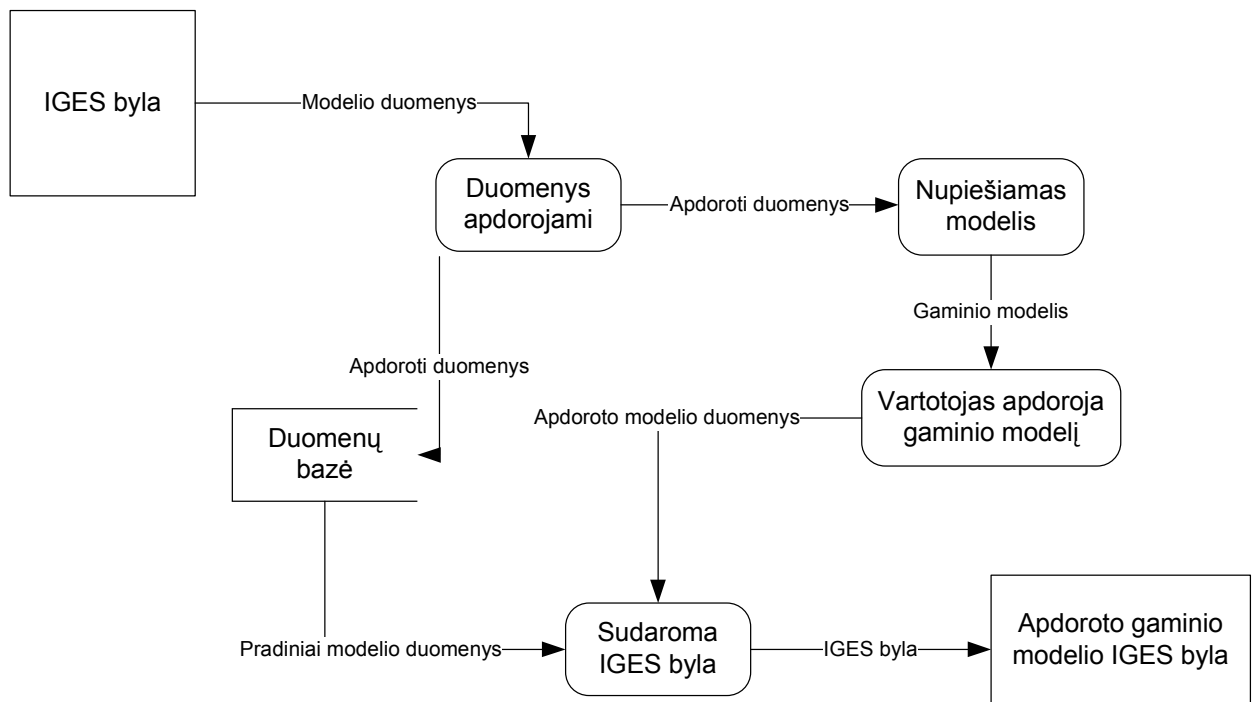
Lentelė	Aprašymas
S_sector	Saugoma visa informacija esanti pirmame iges bylos sektoriuje S
G_sector	Saugoma visa informacija esanti antrajame iges bylos sektoriuje G. Šiame darbe reikalingi šie G sektoriaus parametrai: Parametrų reikšmių skirtukas, Įrašų skirtukas, Modelio erdvės mastelis, Vienetų tipą žyminti vėliavėlė, Minimalus vartotojo ar programinės įrangos užduotas tikslumas, Apytikslė maksimali koordinatės reikšmė
D_sector	Saugoma visa informacija esanti trečiajame iges bylos sektoriuje D. Šiame darbe reikalingi šie D sektoriaus parametrai: Esybės tipo numeris, Parametrų duomenys, Transformacijų matrica, Būsenos numeris, Eilės numeris, Spalvos numeris, Parametrų eilučių kiekis, Formos numeris
P_sector	Saugomas ryšys tarp esybės tipo ir jo eilutės numerio sektoriuje P.
Entity100	Esybės tipo 100 (apskritas lankas) duomenys
Entity110	Esybės tipo 110 (atkarpa) duomenys
Entity116	Esybės tipo 116 (taškas) duomenys
Entity124	Esybės tipo 124 (transformacijų matrica) duomenys
Entity126	Esybės tipo 126 (racionalioji B-splaininė kreivė) papildomi duomenys
Entity126_control	Esybės 126 kontrolinių taškų svoriai
Entity126_points	Esybės 126 kontroliniai taškai
Entity126_vectors	Esybės 126 vektoriai u kryptimi
Entity128	Esybės tipo 128 (Racionalusis B-splaininis paviršius) papildomi duomenys
Entity128_control	Esybės 128 kontrolinių taškų svoriai
Entity128_points	Esybės 128 kontroliniai taškai
Entity128_vector1	Esybės 128 vektoriai u kryptimi
Entity128_vector2	Esybės 128 vektoriai v kryptimi
Entity141	Esybės tipo 128 (Ribojančioji esybė) papildomi duomenys
Entity141_curves	Išorinės iškarpančiosios kreivės duomenys
Entity141_curves2	Vidinės iškarpančiųjų kreivių duomenys
Entity142	Esybės tipo 142 (Kreivė parametriniame paviršiuje) duomenys
Entity143	Esybės tipo 143 (Apribotasis paviršius) duomenys
Entity143_pointers	Iškarpančiųjų kreivių duomenys
Entity144	Esybės tipo 144 (Iškarpytasis (parametrinis) paviršius) duomenys
Entity144_pointers	Vidinių iškarpančiųjų kreivių duomenys

Lentelė 3.2.1: Gaminio modelio duomenų bazės esybės

Kiekvienos esybės parametrai yra aprašyti prieduose, 8.1 skyriuje.

Bendra gaminio modelio duomenų apsikeitimo schema pavaizduota Pav. 3.2.2. Vartotojas programos pagalba atsidaro IGES bylą. Duomenys iš bylos nuskaitomi ir įrašomi į duomenų bazę, tuo pačiu duomenys, reikalingi modelio atvaizdavimui, įrašomi į operatyviąją atminį. Vartotojui koreguojant gaminio modelį, programa su modelio duomenimis atlieka tūkstančius operacijų per sekundę, todėl pagrindiniai duomenys yra išsaugomi operatyviojoje atmintyje. Pagalbiniai duomenys, kurie reikalingi tūrio skaičiavimui, mastelio keitimui ar kitoms operacijoms, nuskaitomi

iš duomenų bazės tik prieš jas vykdant. Todėl neapkraunama sistema ir užtikrinama optimali gaminio modelio redagavimo sparta.



Pav. 3.2.2: Bendra gaminio modelio duomenų apsikeitimo schema pavaizduota

Esant klaidingai *IGES* bylai, vartotojas informuojamas kad byla yra klaidinga ir duomenys nėra nuskaitomi. Programa negali nustatyti ar duomenys yra teisingi ar iškraipyti jeigu *IGES* byla yra struktūriškai tvarkinga. Tokiu atveju programa nuskaitys duomenys ir atvaizduos gaminio modelį pagal nuskaitytus duomenis.

3.2.2. Pagalbiniai duomenys

Pagalbinių duomenų bazė susideda iš vienos lentelės (žr.: Pav. 3.2.3)

default_values	
PK	<u>id</u>
	param_name param_def_value param_value

Pav. 3.2.3: Pagalbinių duomenų DB

param_name – parametro pavadinimas

param_def_value – standartinė parametro reikšmė

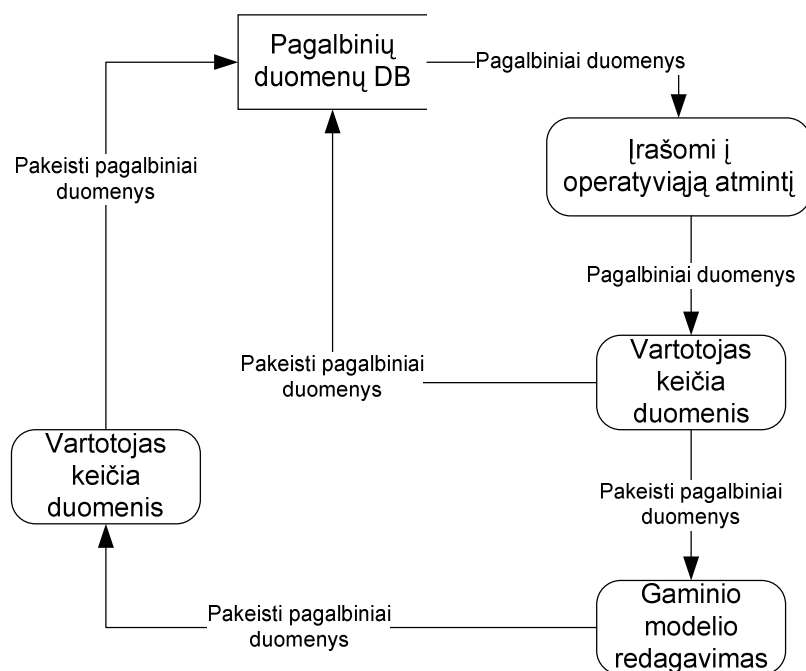
param_value – vartotojo nustatyta parametro reikšmė

Šie parametrai nuskaitomi paleidus programą, todėl ši duomenų bazė yra programos dalis. Kiekvieną kartą vartotojui pakeitus programos parametrus, jie įrašomi į šią duomenų bazę.

Pagalbiniai duomenys:

- **Step size:** diskretizavimo žingsnis. Naudojamas kreivės vertimo ir tūrio skaičiavimo algoritmuose.
- **Angular tolerance:** galima kampinė paklaida. Naudojama algoritmuose su kreivėmis.
- **Distance tolerance:** galima atstumo paklaida. Naudojama kreivės taškų minimizavimo algoritme.
- **Minimum distance:** mažiausias atstumas tarp kontrolinių taškų. Naudojamas kreivės taškų minimizavimo algoritme.
- **Tool diameter:** įrankio diametras. Naudojamas pakavimo algoritme.
- **Packing area:** lapo, kuriame pakuojamos figūros matmenys. Naudojami pakavimo algoritme.
- **Number of areas:** lapų, kuriuose pakuojamos figūros, kiekis. Naudojamas pakavimo algoritme.
- **Time for packing:** pakavimui skirtas laikas. Naudojamas pakavimo algoritme.
- **Undo levels:** išsimintų veiksmų rezultatų kiekis, funkcijų padarinių panaikinimui.

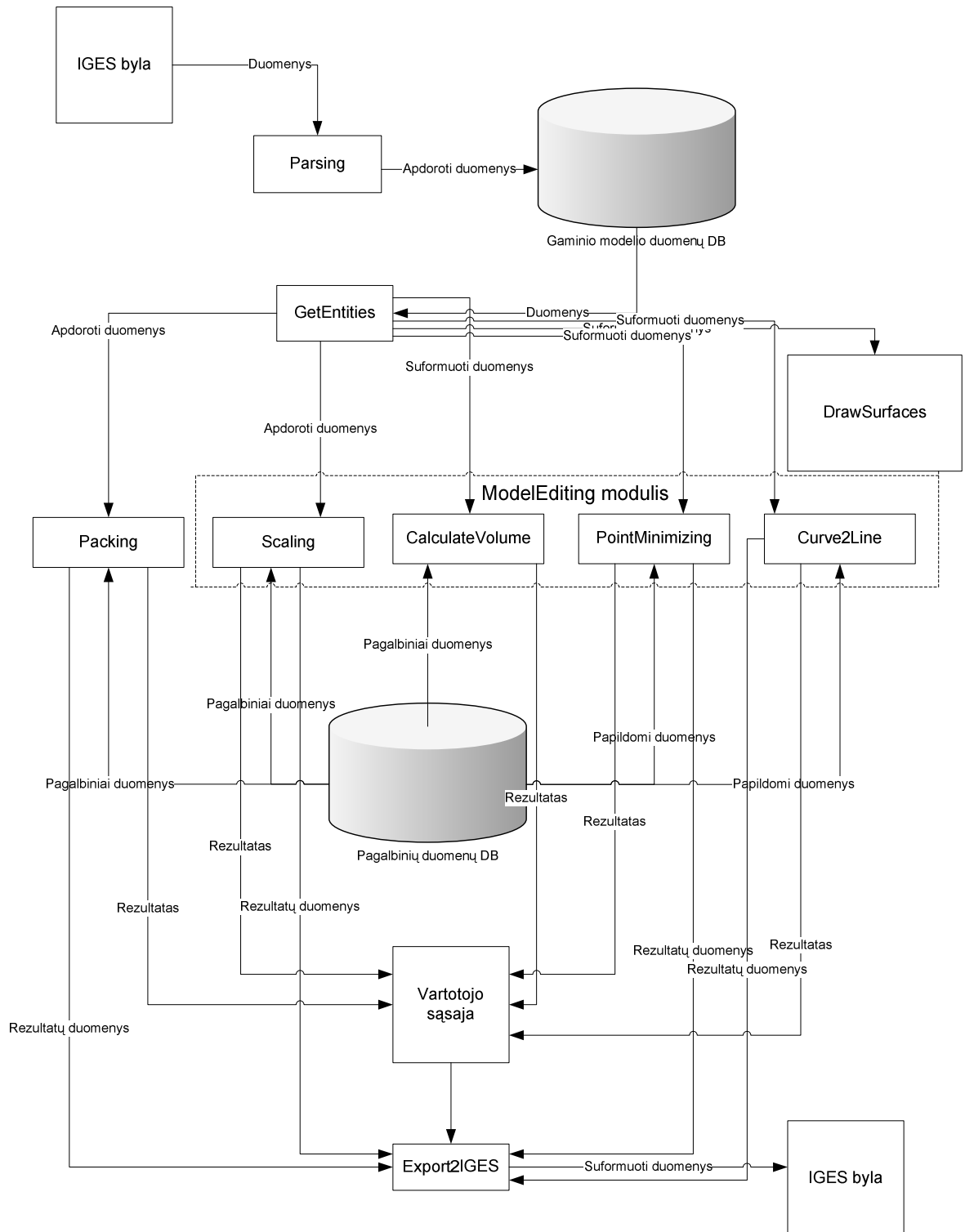
Bendra pagalbinių duomenų apsikeitimo schema pavaizduota *Pav. 3.2.4*.



Pav. 3.2.4: Bendra pagalbinių duomenų apsikeitimo schema

3.3. Sistemos architektūra

Sistemos modulių hierarchinė struktūra pavaizduota *Pav. 3.3.1.*



Pav. 3.3.1: Sistemos modulių hierarchinė struktūra

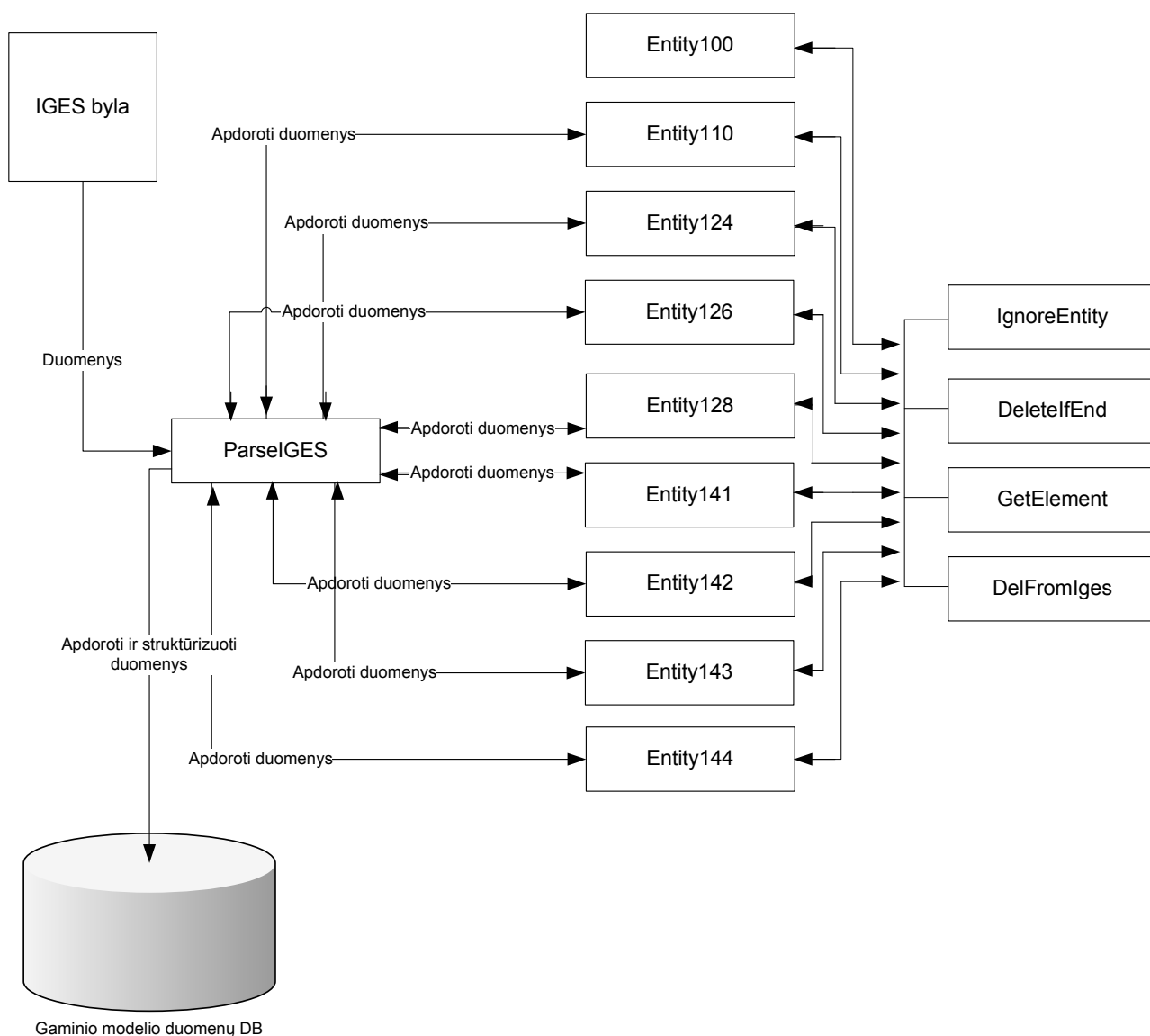
- **Parsing** – apdoroja *IGES* bylą, pasiima reikalingus parametrus ir įrašo į duomenų bazę.

- **GetEntities** – naudoja duomenis iš duomenų bazės ir paruošia gaminio modelį redagavimui.
- **DrawSurface** – vaizduoja gaminio modelį vartotojo sąsajoje.
- **Packing** – realizuoja 2D figūrų pakavimo algoritmą ir vaizduoja rezultatus į ekraną. Vartotojui pasirinkus rezultatai gali būti išsaugoti *IGES* byloje.
- **Scaling** – keičia 3D srities proporcijas ir vaizduoja rezultatus ekrane. Vartotojui pasirinkus rezultatai gali būti išsaugoti *IGES* byloje.
- **CalculateVolume** – vykdo 3D paviršiaus apribotos erdvės tūrio skaičiavimo algoritmą ir rodo rezultatus į ekraną.
- **PointMinimizing** – vykdo 2D figūrų taškų kiekio minimizavimo algoritmą ir išveda rezultatus į ekraną. Vartotojui pasirinkus rezultatai gali būti išsaugoti *IGES* byloje.
- **Curve2Line** – verčia splineinės kreivę į atkarpas bei lankus ir vaizduoja rezultatus į ekraną. Vartotojui pasirinkus rezultatai gali būti išsaugoti *IGES* byloje.

3.4. Programinių modulių ir objektų specifikacijos

3.4.1. „Parsing“ modulis

„Parsing“ modulis atsakingas už duomenų nuskaitymą iš *IGES* bylos. Imami tik tie duomenys kurie bus reikalingi gaminio modelio redagavimui ir optimizavimui. Nuskaityti duomenys išsaugomi duomenų bazėje. „Parsing“ modulio funkcinė hierarchinė struktūra pavaizduota *Pav. 3.4.1.*



Pav. 3.4.1: „Parsing“ modulio funkcinė hierarchinė struktūra

ParseIGES – funkcija kontroliuojanti duomenų nuskaitymą iš *IGES* bylos.

Entity100 - funkcija apdoroja esybės tipą *100*.

Entity110 - funkcija apdoroja esybės tipą *110*.

- Entity116** - funkcija apdoroja esybės tipą *116*.
- Entity124** - funkcija apdoroja esybės tipą *124*.
- Entity126** - funkcija apdoroja esybės tipą *126*.
- Entity128** - funkcija apdoroja esybės tipą *128*.
- Entity141** - funkcija apdoroja esybės tipą *141*.
- Entity142** - funkcija apdoroja esybės tipą *142*.
- Entity143** - funkcija apdoroja esybės tipą *143*.
- Entity144** - funkcija apdoroja esybės tipą *144*.
- IgnoreEntity** – funkcija ignoruoja eilutes esybės kurios programa nepalaiko.
- DeleteIfEnd** – funkcija ištrina eilutės pabaigos simbolius.
- GetElement** – funkcija nuskaitytą vieną elementą, kurį skiria skirtukas.
- DelFromIGES** – papildoma funkcija, ištrina jau nuskaitytą informaciją

„*Parsing*“ modulio algoritmas:

1. Nusiskaitome bylos turinį
2. Kol 73 simbolis lygus *,S‘*
 - 2.1. Nuskaitome reikalingus duomenis
 - 2.2. Ištriname eilutę (82 simbolius)
3. Kol 73 simbolis lygus *,G‘*
 - 3.1. Nuskaitome reikalingus duomenis
 - 3.2. Ištriname eilutę (82 simbolius)
4. Kol 73 simbolis lygus *,D‘*
 - 4.1. Nuskaitome reikalingus duomenis
 - 4.2. Ištriname eilutę (82 simbolius)
5. Kol 73 simbolis lygus *,P‘*
 - 5.1. Nuskaitome pirmąjį skaičių iki skirtuko, tai bus esybės tipas.
 - 5.2. Kreipiamės į atitinkamą funkciją apdorojančią nuskaitytą esybės tipą
 - 5.2.1. Kol nesutinkamas esybių skirtukas
 - 5.2.1.1. Nuskaitome vienos eilutės duomenis ir apdorojam
 - 5.2.1.2. Ištriname eilutę
6. Nuskaitome paskutiniąją *IGES* bylos eilutę – sektorių *,T‘*

3.4.2. „GetEntities“ modulis

„GetEntities“ modulis nuskaito duomenis iš duomenų bazės ir struktūrizuoja pagal esybių tipus. Kiekviena esybė turi savo struktūrą (žr.: *Lentelė 3.4.1*). Esybių tipai 141, 142, 143 ir 144 yra rodyklių rinkinys į 126 tipo esybes, todėl jos savo atskirų struktūrų neturi.

Esybė	Struktūra
100, ZT - Poslinkis ZT ašies kryptimi. X1 - Lanko centro abscisė. Y1 - Lanko centro ordinatė. X2 - Pradžios taško abscisė. Y2 - Pradžios taško ordinatė. X3 - Pabaigos taško abscisė. Y3 - Pabaigos taško ordinatė. aid - esybės identifikacijos numeris. next - nuoroda į sekančios esybės struktūrą prev - nuoroda į buvusios esybės struktūrą	struct Figure100 { float Z,X0,Y0,X1,Y1,X2,Y2; int aid; Figure100* next; Figure100* prev; };
110, X1 - Pradžios taško P1 koordinatė 0X ašyje. Y1 - Pradžios taško P1 koordinatė 0Y ašyje. Z1 - Pradžios taško P1 koordinatė 0Z ašyje. X2 - Pabaigos taško P2 koordinatė 0X ašyje. Y2 - Pabaigos taško P2 koordinatė 0Y ašyje. Z2 - Pabaigos taško P2 koordinatė 0Z ašyje. aid - esybės identifikacijos numeris. next - nuoroda į sekančios esybės struktūrą prev - nuoroda į buvusios esybės struktūrą	struct Figure110 { float X1,Y1,Z1,X2,Y2,Z2; int aid; Figure110* next; Figure110* prev; };
124, R11 - Viršutinė R matricos eilutė. R12 - ... R13 - ... T1 - Viršutinis T vektoriaus narys. R21 - Vidurinė R matricos eilutė. R22 - ... R23 - ... T2 - Vidurinė T vektoriaus narys. R31 - Apatinė R matricos eilutė. R32 - ... R33 - ... T3 - Apatinis T vektoriaus narys. aid - esybės identifikacijos numeris. next - nuoroda į sekančios esybės struktūrą prev - nuoroda į buvusios esybės struktūrą	struct Figure124 { float R11,R12,R13,R21,R22,R23,R31,R32,R33,T1,T2,T3; int aid; Figure124* next; Figure124* prev; };
126, KnotCount - mazgų vektoriaus reikšmių skaičius. Order aaa PointCount – kontrolinių taškų kiekis Knots – mazgų vektorių masyvas Points – kontrolinių taškų masyvas minX – minimali kontrolinių taškų X koordinatė minY – minimali kontrolinių taškų Y koordinatė minZ – minimali kontrolinių taškų Z koordinatė maxX – maksimali kontrolinių taškų X koordinatė maxY – maksimali kontrolinių taškų Y koordinatė maxZ – maksimali kontrolinių taškų Z koordinatė aid - esybės identifikacijos numeris. next - nuoroda į sekančios esybės struktūrą prev - nuoroda į buvusios esybės struktūrą	struct Figure126 { int KnotCount,Stride,Order,PointCount; float Knots[MAXVECTORSV]; float points[MAXPOINTS]; float minX,minY,minZ,maxX,maxY,maxZ; int aid,linenumber; Figure126* next; Figure126* prev; Figure126* in; };

<p>in – nuoroda į pirmąją vidinę iškarpančiąją kreivę 128, UKnotCount - mazgų vektoriaus u kryptimi reikšmių skaičius. VKnotCount - mazgų vektoriaus v kryptimi reikšmių skaičius. UOrder – bazines u funkcijos laipsnis +1 VOrder - bazines v funkcijos laipsnis +1 PointCountU - kontrolinių taškų kiekis u kryptimi PointCountV - kontrolinių taškų kiekis v kryptimi UKnots – mazgų vektorių masyvas u kryptimi VKnots – mazgų vektorių masyvas v kryptimi Points – kontrolinių taškų masyvas rpoints – realių taškų masyvas linenumber – eilutės numeris P sektoriuje M1 – bazinės funkcijos u kryptimi laipsnis M2 – bazinės funkcijos v kryptimi laipsnis minX – minimali kontrolinių taškų X koordinatė minY – minimali kontrolinių taškų Y koordinatė minZ – minimali kontrolinių taškų Z koordinatė maxX – maksimali kontrolinių taškų X koordinatė maxY – maksimali kontrolinių taškų Y koordinatė maxZ – maksimali kontrolinių taškų Z koordinatė aid - esybės identifikacijos numeris. next - nuoroda į sekančios esybės struktūrą prev - nuoroda į buvusios esybės struktūrą out – nuoroda į išorinę iškarpančiąją kreivę</p>	<pre> struct Figure128 { int UKnotCount, VKnotCount, UStride, VStride, UOrder, VOrder, PointCountU, PointCountV; float VKnots[MAXVECTORSV], UKnots[MAXVECTORSU]; float points[MAXPOINTS]; float rpoints[MAXPOINTS]; float minX, minY, minZ, maxX, maxY, maxZ; int aid, linenumber, M1, M2; Figure128* next; Figure128* prev; Figure126* out; }; </pre>
---	--

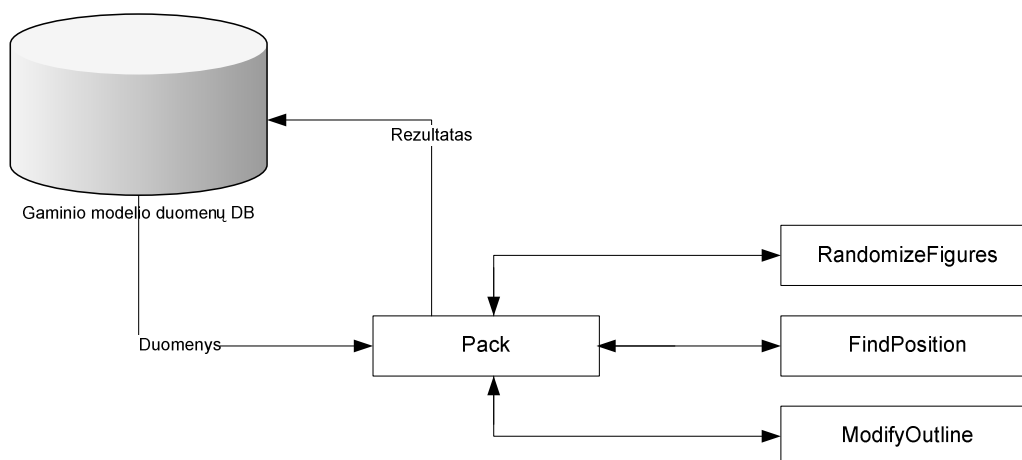
Lentelė 3.4.1: Esybių tipų struktūros

3.4.3. „DrawSurface“ modulis

„DrawSurface“ modulis piešia gaminio modelį grafinėje vartotojo sąsajoje. Prieš kreipiantis į šį modulį, visa informacija nuskaityta į operatyviają atmintį.

3.4.4. „Packing“ modulis

„Packing“ modulio funkcinė hierarchinė struktūra pavaizduota Pav. 3.4.2.



Pav. 3.4.2: „Packing“ modulio funkcinė hierarchinė struktūra

Pack – funkcija kontroliuojanti pakavimo algoritmą.

RandomizeFigures – išmaišo figūrų eiliškumą.

FindPosition – keičia figūros orientaciją koordinacių sistemos atžvilgiu.

ModifyOutline – atnaujina dengiantį kontūrą.

Figūros duomenų struktūra pavaizduota *Lentelė 3.4.2*.

Struktūra	Aprašymas
<pre>struct figures { int Mas[MAXFIG][2]; inner_plot* inners; figures* next; figures* prev; };</pre>	Figūros struktūra. Mas – figūros taškų masyvas inners – nuoroda į „skylių“ struktūrą next – nuoroda į sekančią figūrą prev – nuoroda į ankstesnę figūrą
<pre>struct inner_plot { int Mas[MAXINNER][2]; inner_plot* next; inner_plot* prev; };</pre>	Figūros „skylių“ struktūra. Mas – figūros „skylių“ taškų masyvas next – nuoroda į sekančią figūros „skylę“ prev – nuoroda į ankstesnę figūros „skylę“

Lentelė 3.4.2: Figūros duomenų struktūra

„Packing“ modulio algoritmas aprašytas 2.10 skyriuje

3.4.5. „ModelEditing“ modulis

„ModelEditing“ modulis susideda iš 4 pagrindinių funkcijų: „Scaling“, „CalculateVolume“, „PointMinimizing“, „Curve2Line“.

Visos šios funkcijos naudoja jau struktūrizuotus duomenis aprašytus 3.4.2 skyrelyje.

„Scaling“ funkcija keičia gaminio modelio proporcijas, pasirinktoje srityje. Gauti rezultatai išsaugomi duomenų bazėje bei vaizduojami ekrane. Šio modulio algoritmas aprašytas 2.9 skyriuje.

„CalculateVolume“ funkcija apskaičiuoja pasirinktų paviršių ribojamą tūrį. Gauti rezultatai išsaugomi duomenų bazėje bei rodomi ekrane. Šio modulio algoritmas aprašytas 2.8 skyriuje.

„PointMinimizing“ funkcija sumažina kreivės kontrolinių taškų kiekį, atsižvelgiant į vartotojo nurodytus parametrus. Gauti rezultatai išsaugomi duomenų bazėje bei vaizduojami ekrane. Šio modulio algoritmas aprašytas 2.11 skyriuje.

„Curve2Line“ funkcija verčia vartotojo pasirinktą kreivę į taškus pagal vartotojo nustatytus parametrus. Gauti rezultatai išsaugomi duomenų bazėje bei vaizduojami ekrane. Šio modulio algoritmas aprašytas 2.11 skyriuje.

3.5. Testavimo medžiaga

Šiame skyriuje pateikiami realizuotų algoritmų testavimo rezultatai.

3.5.1. Duomenų iš IGES bylos skaitymas ir rašymas

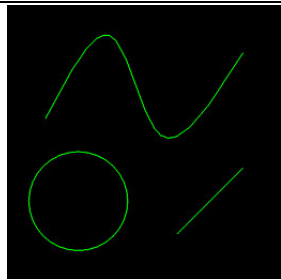
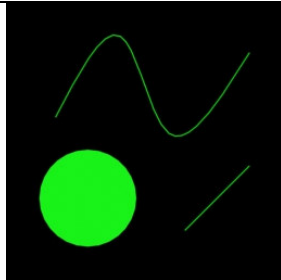
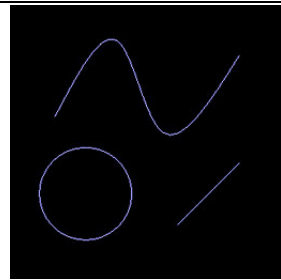
Išorinė PĮ	<i>Assistant3D</i> skaito išorinės PĮ sugeneruotą <i>IGES</i> bylą	Išorinė PĮ skaito <i>Assistant3D</i> sugeneruotą <i>IGES</i> bylą
<i>3D Studio Max R 5.0</i>	taip	taip
<i>MasterCAM v 6.0</i>	taip	taip
<i>MasterCAM v 9.0</i>	taip	taip

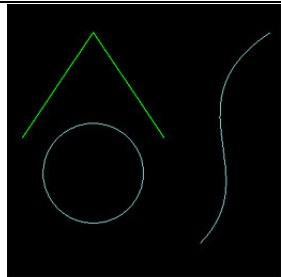
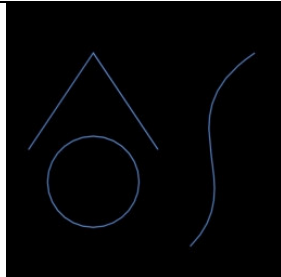
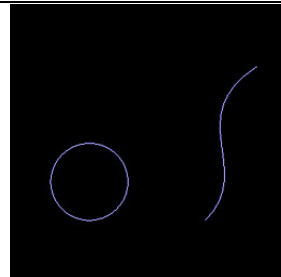
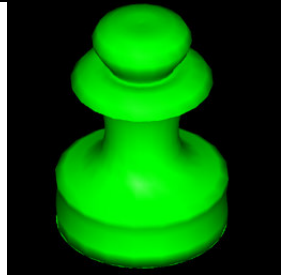
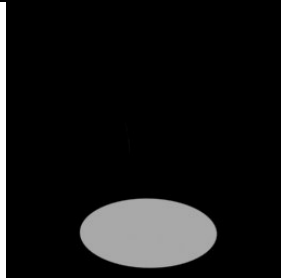
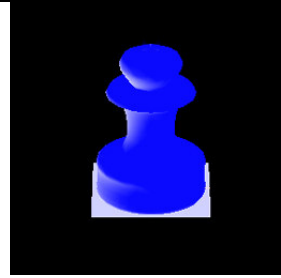
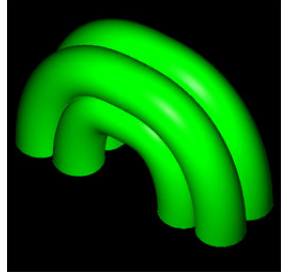

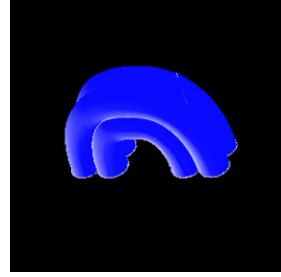
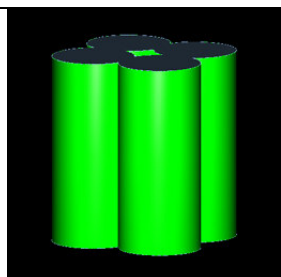
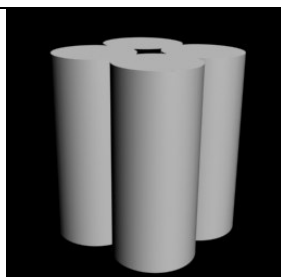
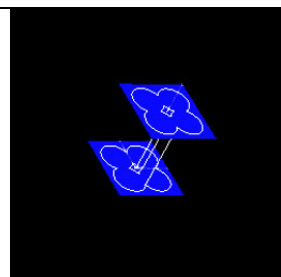
Lentelė 3.5.1: Duomenų mainai tarp CAD/CAM sistemų ir *Assistant3D*, naudojant *IGES* bylas

Sukurtos programinės įrangos sugebėjimas bendrauti su CAD/CAM sistemomis, naudojant *IGES* formato bylas duomenų pernešimui, analizuojamas, tikrinant sėkmingo duomenų perėmimo faktą (žr.: Lentelė 3.5.1). Duomenų perėmimo ir perdavimo korektiškumas testuojamas 3.5.2 skyrelyje.

Išvada: Duomenų mainai tarp programinių įrangų įmanomi, tačiau perimtų duomenų korektiškumas dar nepatvirtintas.

3.5.2. IGES bylos turinio interpretavimas ir vaizdavimas

<i>MasterCAM v 9.0</i>	<i>3D Studio Max R 5.0</i>	<i>Assistant3D</i>	Bylos turinys
			003_line-arc-spline_cam.IGS 100 tipas: lankas 110 tipas: atkarpa 124 tipas: transformacijų matrica 126 tipas: NURBS kreivė
Komentaras: Bylą sugeneravo <i>MasterCAM</i> sistema. Visi duomenys yra aprašyti dvimatėje erdvėje. <i>3D Studio Max</i> visas esybes vaizduoja teisingai, išskyrus apskritimą(lanką), kurį pagal nutylėjimą užpildo. Informacija neprarandama.			

			<p>004_line-arc-spline_max.IGS</p> <p>100 tipas: lankas 106 tipas: 2D laužtė 124 tipas: transformacijų matrica 126 tipas: NURBS kreivė 314 tipas: spalvos esybė</p>
<p>Komentaras: Bylą sugeneravo <i>3D Studio Max</i>. Visi duomenys yra aprašyti dvimatėje erdvėje. <i>MasterCAM</i> informacija vaizduoja teisingai. Pakartotinai bylą atidarius su <i>3D Studio Max</i> sistema, apskritimas vėlgi užpildomas. Naudojama 106 tipo esybė, kuri šiame darbe nenagrinėjama.</p>			
			<p>006_pawn_cam.IGS</p> <p>116 tipas: taškas 126 tipas: NURBS kreivė 128 tipas: NURBS paviršius 142 tipas: kreivė par. paviršiuje 144 tipas: iškarpytasis paviršius 314 tipas: spalvos esybė</p>
<p>Komentaras: Bylą sugeneravo <i>MasterCAM</i> sistema. Naudojamas NURBS paviršius ir iškarpytasis paviršius. <i>3D Studio Max</i> nepavyksta užkrauti NURBS paviršiaus ir vaizduojamas tik iškarpytasis paviršius. neiškerpa iškarpytojo paviršiaus. Prarandami duomenys.</p>			
			<p>001_4cyl-bend_max.IGS</p> <p>116 tipas: taškas 126 tipas: NURBS kreivė 128 tipas: NURBS paviršius 141 tipas: kreivė par. paviršiuje 143 tipas: iškarpytasis paviršius 314 tipas: spalvos esybė</p>
<p>Komentaras: Bylą sugeneravo <i>3D Studio Max</i>. Naudojami NURBS paviršiai. Duomenys interpretuojami teisingai visų sistemų. Informacija neprarandama.</p>			
			<p>002_4cylinders_max.IGS</p> <p>116 tipas: taškas 126 tipas: NURBS kreivė 128 tipas: NURBS paviršius 141 tipas: kreivė par. paviršiuje 143 tipas: iškarpytasis paviršius 314 tipas: spalvos esybė</p>
<p>Komentaras: Bylą sugeneravo <i>3DS Max</i>. Naudojamas NURBS ir iškarpytieji paviršiai. <i>MasterCAM</i> nepavyko tiesiogiai panaudoti užkrautų duomenų, iškarpytųjų paviršių vaizdavimui. Tačiau jie buvo sugeneruoti, papildomai apdorojus duomenis. <i>Assistant3D</i> nerodo ir neiškarpo paviršių.</p>			

Lentelė 3.5.2: Duomenų interpretavimo korektiškumas

Duomenų interpretavimo korektiškumo testas (žr.: *Lentelė 3.5.2*) parodė, kad skirtingų sistemų sugeneruotos bylos, interpretuojamos skirtingai. Sistema, sugeneravusi *IGES* bylą, ją interpretuoja nepriekaištingai, tačiau kitų sistemų pateiktos interpretacijos kartais skiriasi.

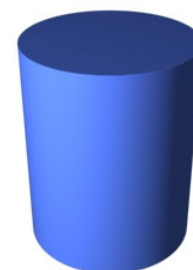
Išvada: Visos įdiegti esybių tipai interpretuojamos gerai, išskyrus iškarpytuosius paviršius. Dažnai iškarpančiosios kreivės išeina iš parametrų erdvės arba jų projekcijos iškarpytajame paviršiuje neatitinka iškarpančiosios kreivės proporcijų modelio erdvėje. Taip pat kartais (retai) neteisingai interpretuojamos kitų tipų esybės. Ši sritis reikalauja gilesnės analizės.

3.5.3. Tūrio skaičiavimas

Skaičiuojamas modelio #01 paviršiais apribotas tūris (žr.: *Pav. 3.5.1*). Teoriškai apskaičiuotas tūris $V_{teor} = 62832 \text{ kv. vnt.}$

Naudojant parametą $d = 0,1$ apskaičiuotas $V = 62646 \text{ kv. vnt.}$ Paklaida siekia 0,3%. Skaičiavimai trunka 0,5 sekundės.

Naudojant parametą $d = 0,01$ apskaičiuotas $V = 62819 \text{ kv. vnt.}$ Paklaida siekia 0,02%. Skaičiavimai trunka 15 sekundžių.



Pav. 3.5.1: Modelio #01 vaizdas

Skaičiuojamas modelio #02 paviršiais apribotas tūris (žr.: *Pav. 3.5.2*). Teoriškai apskaičiuotas tūris $V_{teor} \approx 1733000 \text{ kv. vnt.}$

Naudojant parametą $d = 0,1$ apskaičiuotas $V = 1717582 \text{ kv. vnt.}$ Paklaida siekia 0,9%.

Naudojant parametą $d = 0,001$ apskaičiuotas $V = 1729542 \text{ kv. vnt.}$ Paklaida siekia 0,02%.

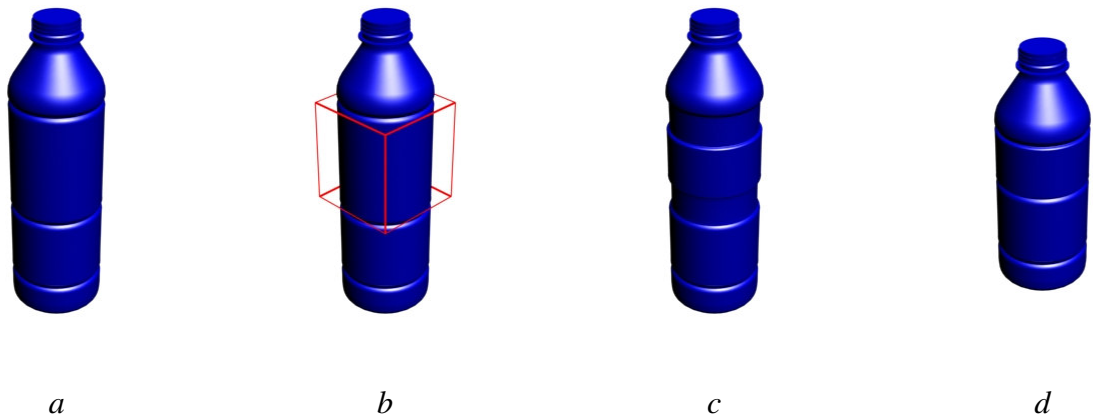


Pav. 3.5.2: Modelio #02 vaizdas

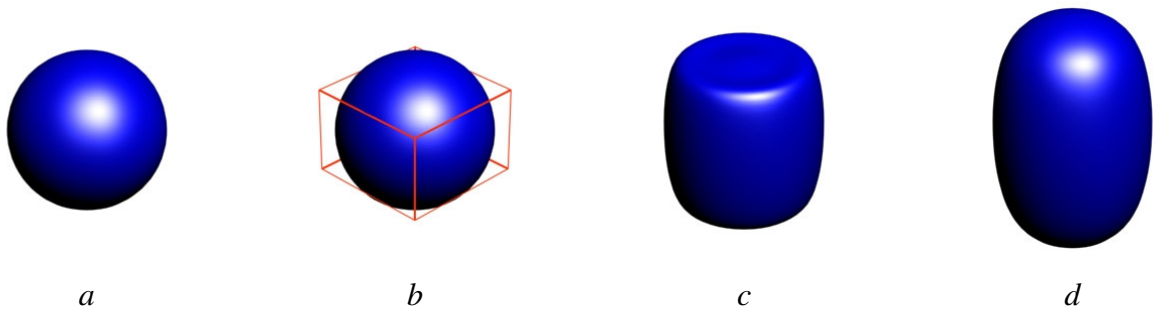
Algoritmas, naudojant numatytus atvejus, elgiasi kaip ir tikėtasi. Dėl to, kad visada skaičiuojamas į uždara paviršiaus kreivę įbrėžto daugiakampio plotas (žr. 2.8 skyrius), galutinis rezultatas visada būna mažesnis už teorinį. Modelio #02 tūrio skaičiavimo atveju paklaidos turėtų būti mažesnės, dėl vidinių ir išorinių paviršių diskretizavimo paklaidos kompensacijos. Tačiau to negalima nustatyti, jei teorinis tūris nėra tikslus.

Išvada: Eksperimentų dėka galima nustatyti ir patvirtinti tikėtiną paklaidą, priklausančią nuo užduoto žingsnio. Atsižvelgiant į tai, kad apskaičiuotas tūris visada mažesnis už teorinį (tikrąjį), rezultatą padauginus iš tikėtiną paklaidą atitinkančio koeficiento, būtų galima sumažinti žingsnių skaičių ir padidinti skaičiavimo tikslumą.

3.5.4. Pasirinktos modelio srities proporcijų koregavimas



Pav. 3.5.3: Modelio #03 vaizdas: a) prieš transformaciją; b) raudonai apibrėžta transformuojama sritis A; c) srities A mastelis 0,5 išilgai Oy ašies, susietoji sritis B tuščia; d) A srities mastelis 0,5 išilgai Oy ašies, $B = \overline{A}$.



Pav. 3.5.4: Modelio #04 vaizdas: a) prieš transformaciją; b) raudonai apibrėžta transformuojama sritis A; c) srities A mastelis 2,0 išilgai Oy ašies, susietoji sritis B tuščia; d) A srities mastelis 2,0 išilgai Oy ašies, $B = \overline{A}$.

Pav. 3.5.3 ir Pav. 3.5.4 vaizduojama modelio reakcija į pasirinktos srities proporcijų keitimą (žr.: 2.9 skyrius). Modelio srities dalis Pav. 3.5.3.b ir Pav. 3.5.4.b transformuojama išilgai Oy ašies. Nenaudojant susietosios srities B rezultatas gali būti nepageidautinas (žr.: Pav. 3.5.3.c ir Pav. 3.5.4.c), dėl netransformuojamos ir transformuojamos sričių padėčių santykio, viena kitos atžvilgiu, išardymo. Tokiu atveju papildomai naudojama susietoji sritis B (žr.: Pav. 3.5.3.d ir Pav. 3.5.4.d), išlaikanti padėties erdvėje santykį tarp A ir B.

Išvada: Srities A proporcijų koregavimas įtakoja kontrolinius taškus, esančius tik srities A ribose (skaitant, kad sritis B tuščia). Realizavus srities A gradientiškai mažėjantį poveikį erdvėje, už srities ribų, priklausomą nuo konkrečios funkcijos, būtų galima pasiekti naudingų rezultatų, suteikiančių modelio koregavimui daugiau lankstumo.

3.5.5. Pakavimas dvimatėje erdvėje

Dėl šio algoritmo sudėtingumo ir daugialypiškumo, nepavyko jo galutinai suderinti. Pateikti tik preliminarūs rezultatai (žr.: *Pav. 3.5.5*), kai $d = 5$, pakavimo laikas $t = 5 \text{ min}$.



Pav. 3.5.5: Pakavimas dvimatėje erdvėje

Išvada: Algoritmas privalo būti papildomai studijuojamas ir pakartotinai realizuojamas, siekiant rezultatų, kurie galėtų varžytis su jau egzistuojančiais panašiais algoritmais.

3.5.6. NURBS kreivės koregavimas

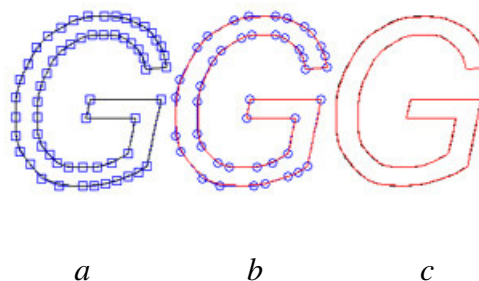
Kontrolinių taškų minimizavimas

Analizuojama programinio paketo *CorelDraw* sugeneruota NURBS kreivė su 96 kontroliniais taškais (žr.: Pav. 3.5.6.a). Kreivės matmenys 25×25 vnt. Gauti rezultatai, apdorojus kreivę kontrolinių taškų minimizavimo algoritmu su įvairiomis parametru reikšmėmis, pateikti Pav. 3.5.6 - Pav. 3.5.9. Eksperimento metu buvo priimta, kad atstumo paklaidos reikšmė ε yra pastovi ir nekeičiama.

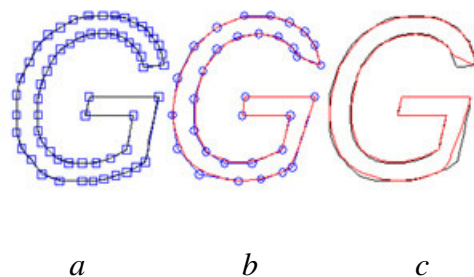
Pav. 3.5.6 pateiktas algoritmo rezultatas, kai minimalus atstumas tarp kontrolinių taškų $d = 2$, o galima kampinė paklaida $\gamma = 1^\circ$. Buvo pašalinti 32 kontroliniai taškai, tai sudaro 33% atminties resursų, skirtų informacijos apie kreivę saugojimui. Kreivės forma po optimizavimo visiškai nepakito.

Pav. 3.5.7 pateiktas algoritmo rezultatas, kai minimalus atstumas tarp kontrolinių taškų $d = 8$, o galima kampinė paklaida $\gamma = 1^\circ$. Buvo pašalintas 41 kontrolinis taškas, tai sudaro 43% atminties resursų, skirtų informacijos apie kreivę saugojimui. Optimizavimo metu buvo pašalinti kai kurie svarbūs taškai, tarp kurių atstumas buvo mažesnis nei d reikšmė. Tai stipriai įtakojo kreivės iškraipymą.

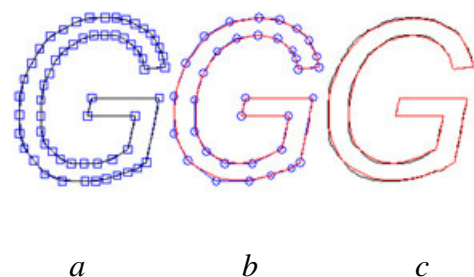
Pav. 3.5.8 pateiktas algoritmo rezultatas, kai minimalus atstumas tarp kontrolinių taškų $d = 2$, o galima kampinė paklaida $\gamma = 30^\circ$. Buvo pašalinti 33 kontroliniai taškai, tai sudaro 34% atminties resursų, skirtų informacijos apie kreivę saugojimui. Optimizuota kreivė nežymiai skiriasi nuo originalo. Rezultatas patenkinamas.



Pav. 3.5.6: NURBS kreivės kontrolinių taškų minimizavimas: a) kontroliniai taškai prieš operaciją; b) kontroliniai taškai po operacijos c) kreivių palyginimas

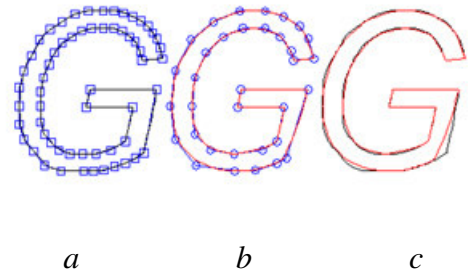


Pav. 3.5.7: NURBS kreivės kontrolinių taškų minimizavimas: a) kontroliniai taškai prieš operaciją; b) kontroliniai taškai po operacijos c) kreivių palyginimas



Pav. 3.5.8: NURBS kreivės kontrolinių taškų minimizavimas: a) kontroliniai taškai prieš operaciją; b) kontroliniai taškai po operacijos c) kreivių palyginimas

Pav. 3.5.9 pateiktas algoritmo rezultatas, kai minimalus atstumas tarp kontrolinių taškų $d = 5$, o galima kampinė paklaida $\gamma = 30^\circ$. Buvo pašalinti 36 kontroliniai taškai, tai sudaro 37,5% atminties resursų, skirtų informacijos apie kreivę saugojimui. Rezultatas geresnis, nei *Pav. 3.5.7*, tačiau buvo panaikinti trys svarbūs, formuojantys kampą, taškai. Dėl to kreivė iškraipyta ir toks rezultatas nėra patenkinamas.



Pav. 3.5.9: NURBS kreivės kontrolinių taškų minimizavimas: a) kontroliniai taškai prieš operaciją; b) kontroliniai taškai po operacijos c) kreivių palyginimas

Iš testavimo rezultatų matyti, kad didelę reikšmę optimizuotos kreivės korektiškumui turi minimalaus atstumo tarp kontrolinių taškų parametras. Geriausius rezultatus duoda galimos kampo paklaidos reikšmės didinimas (iki tam tikros ribos). Eksperimentuojant konkrečiomis kiekvienai kreivei parametrų reikšmėmis galima gan kompaktiškai suglaudinti informaciją apie kreivę, jos neiškraipant.

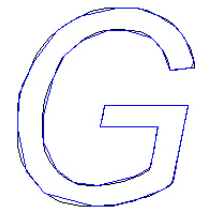
Išvada: Išsamiau paanalizavus *NURBS* kreivių charakteristikas ir įdiegus į algoritmą svorių bei mazgų vektoriaus koregavimo veiksmus, galima būtų padidinti algoritmo efektyvumą bent dvigubai.

Kreivės vertimas atkarpomis ir lankais

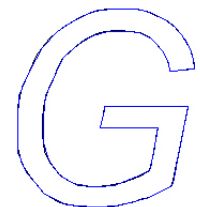
Ta pati kreivė (žr.: *Pav. 3.5.6.a*) verčiama į atkarpas, naudojant 2.11 skyriuje aprašytą algoritmą. Galima kampinė paklaida priskirta $\gamma = 3^\circ$, šiame teste ji laikoma pastovia ir nekintančia.

Pav. 3.5.10 juodai pavaizduota kreivė atitinka originalią kreivę, mėlynos linijos žymi atkarpas, į kurias kreivė buvo suskaidyta. Bandymas buvo atliktas su diskretizavimo žingsniu $d = 2$.

Pav. 3.5.11 iliustruoja tą pačią situaciją, tačiau diskretizavimo žingsnis $d = 1$. Šio atveju diskretizavimo procesas daug tikslesnis, tačiau sunaudojama dvigubai daugiau procesoriaus laiko ir atminties resursų.

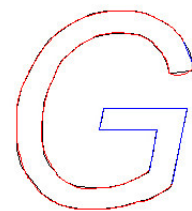


Pav. 3.5.10: NURBS kreivės vertimas į atkarpas



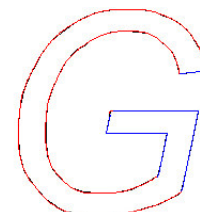
Pav. 3.5.11: NURBS kreivės vertimas į atkarpas

Pav. 3.5.12 vaizduoja tos pačios kreivės vertimą į atkarpas bei lankus. Lankai pažymėti raudonai. Diskretizavimo žingsnis $d = 2$. Pastebimi nepageidautini lankų fragmentai.



Pav. 3.5.12: NURBS kreivės vertimas į atkarpas ir lankus

Pav. 3.5.13 iliustruoja kreivės vertimą į atkarpas bei lankus, kai diskretizavimo žingsnis $d = 1$. Sugeneruotos esybės beveik idealiai padengia originalų kontūrą. Atkarpų atveju, taikant didesnio tikslumo cenzą, „išlystų“ didesnės paklaidos, nei atkarpų su lankais atveju.



Pav. 3.5.13: NURBS kreivės vertimas į atkarpas ir lankus

Išvada: Algoritmas veikia korektiškai, tačiau dar galima padidinti jo efektyvumą reguliuojant atkarpų santykio cenzą, sprendžiant ar naudoti atkarpą ar lanką. Taip pat eksperimentų būdu galima nustatyti pačias efektyviausias parametrų reikšmes.

4. VARTOTOJO DOKUMENTACIJA

4.1. Funkcinis sistemos „Assistant3D“ aprašymas

Programinė įranga „Assistant3D“ skirta gaminio modelio apdorojimui. Gaminio modelio vadinamas gamini aprašantis geometrinis modelis, naudojamas CAM (*Computer Aided Manufacturing*) sistemų, kad realizuoti jo kompiuterizuotą gamybą.

Reikalavimai duomenims:

- Programinė įranga skaito ir rašo duomenis į *IGES* formato bylas.
- Palaikomos geometrinės esybės: lankas, atkarpa, *NURBS* kreivė, *NURBS* paviršius, iškarpytasis *NURBS* paviršius.
- Duomenų mainai testuoti su *MasterCAM* ir *3D Studio Max* sistemomis.

Apdorojimo funkcijos:

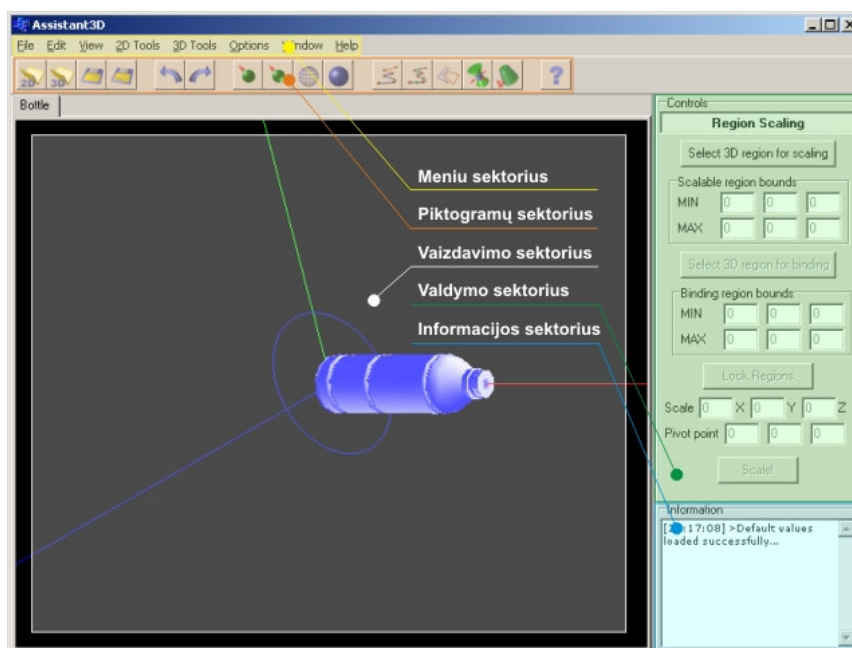
- *NURBS* kreivės vertimas lankais bei atkarpomis.
- *NURBS* kreivės kontrolinių taškų minimizavimas.
- Dvimačių figūrų pakavimas.
- Pasirinktos trimatės srities proporcijų keitimas.
- *NURBS* paviršiais apribotos erdvės tūrio skaičiavimas.

4.2. Sistemos „Assistant3D“ vadovas

Šiame skyriuje aprašomos programinės įrangos „3D Assistant“ funkcijos, klaidų pranešimai ir rekomenduojami įrankių naudojimo būdai, siekiant efektyviausio rezultato.

4.2.1. Vartotojo sąsaja

Grafinė vartotojo sąsaja padalinta į penkis sektorius (žr.: Pav. 4.2.1).



Pav. 4.2.1: Programos „Assistant3D“ grafinė vartotojo sąsaja






Kiekvienas iš sektorių turi specifinę paskirtį:





- **Menu sektorius:** visos „Assistant3D“ funkcijos gali būti inicijuojamos šiame sektoriuje. Meniu yra dviejų lygių. Viršutiniame lygyje gali būti pasirinktos funkcijų kategorijos (Pvz.: *file*, *view*), o apatiniame kategorijai priklausančios funkcijos (Pvz.: *open*, *redo*).
- **Piktogramų sektorius:** šiame sektoriuje gali būti inicijuojamos pagrindinės ir dažniausiai naudojamos funkcijos. Jos atitinka funkcijas iš meniu sektoriaus, bet yra lengviau pasiekiamos (žr.: 4.2.2 skyrelis).
- **Vaizdavimo sektorius:** sektorius skirti aktyvios bylos turinio vaizdavimui. Objektai gali būti vaizduojant bet kokioje projekcijoje, bet koku rakursu. Stebėtojo pozicija gali būti keičiama, naudojant pelės mygtukus: vidurinisys – traukti vaizdą skersai arba išilgai ekrano (klaviatūros klavišas „P“), raktukas –






artinti/tolinti (klaviatūros klavišas „Z“), dešinysis – sukti vaizdą. Kairiuoju mygtuku pasirenkamos esybės ar sritys, jei to reikalauja inicijuota funkcija.

- **Valdymo sektorius:** visos valdymo reikalaujančios funkcijos valdomos šiame sektoriuje (žr.: 4.2.3 skyrelis). Funkcijos valdomos, užduodant atitinkamas parametrų reikšmes ir spaudžiant konkrečius veiksmus inicijuojančius mygtukus.
- **Informacijos sektorius:** visa sistemos generuojama informacija pateikiama šiame sektoriuje.

4.2.2. Funkcijų sąrašas

Nr.	Funkcija	Inicijavimo kelias	Pikt.	Aprašymas
1.1	Atidaryti 2D bylą	<i>File</i> → <i>Open 2D</i>		Leidžia pasirinkti <i>IGES</i> formato bylą, esančia kompiuterio kietajame diske. Programa užkraus atkarpas, apskritimus ir modelio erdvės kreives. Atidaryta byla skelbiama aktyvia.
1.2	Atidaryti 3D bylą	<i>File</i> → <i>Open 3D</i>		Leidžia pasirinkti <i>IGES</i> formato bylą, esančia kompiuterio kietajame diske. Programa užkraus paviršius, iškarpytuosius paviršius ir iškarpančiąsias kreives parametrinėje erdvėje. Atidaryta byla skelbiama aktyvia.
1.3	Išsaugoti kaip...	<i>File</i> → <i>Save As...</i>		Leidžia pasirinkti <i>IGES</i> formato bylos pavadinimą ir vietą kietajame diske, kur bus išsaugoti duomenys. Ši byla bus paskelbta aktyvia.
1.4	Išsaugoti	<i>File</i> → <i>Save</i>		Išsaugo duomenis aktyvioje byloje.
1.5	Uždaryti	<i>File</i> → <i>Close</i>		Uždaro aktyvią bylą, tačiau neišeina iš programos.
1.6	Išeiti	<i>File</i> → <i>Exit</i>		Uždaro „Assistant3D“ programą.
2.1	Panaikinti paskutinės funkcijos	<i>Edit</i> → <i>Undo</i>		Panaikina paskutinės funkcijos darbo rezultatus, jei rezultatas vartotojo netenkina. Jeigu po funkcijos A rezultato panaikinimo nebuvo inicijuota kita

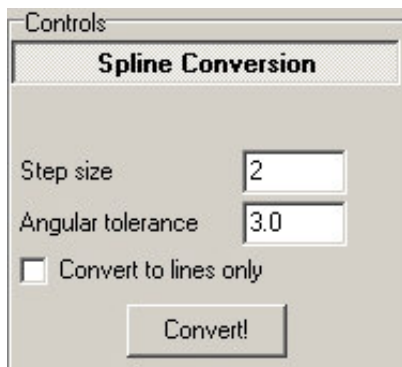
	padarinius			funkcija, tada A rezultatus galima gražinti, naudojant 2.2 funkciją.
2.2	Gražinti panaikintus padarinius	<i>Edit → Redo</i>		Gražina panaikintus (naudojant 2.1 funkciją) paskutinės funkcijos darbo rezultatus.
2.3	Kopijuoti tekstą į atmintį	<i>Edit → Copy Text</i>		Kopijuoja pažymėtą tekstą iš aktyvaus lauko į atmintį.
2.4	Įdėti tekstą	<i>Edit → Paste Text</i>		Įkelia nukopijuotą tekstą iš atminties į aktyvų lauką.
3.1	Vaizdas iš viršaus/apačios	<i>View → Top/Bottom</i>		Aktyvuoja projekciją iš viršaus programos vaizdavimo sektoriuje. Dar kartą inicijavus funkciją, bus aktyvuota projekcija iš apačios.
3.2	Vaizdas iš kairės/dešinės	<i>View → Left/Right</i>		Aktyvuoja projekciją iš kairės programos vaizdavimo sektoriuje. Dar kartą inicijavus funkciją, bus aktyvuota projekcija iš dešinės.
3.3	Vaizdas iš priekio/galo	<i>View → Front/Back</i>		Aktyvuoja projekciją iš priekio programos vaizdavimo sektoriuje. Dar kartą inicijavus funkciją, bus aktyvuota projekcija iš galo.
3.4	Vaizdas iš perspektyvos	<i>View → Perspective</i>		Aktyvuoja perspektyvos projekciją programos vaizdavimo sektoriuje.
3.5	Rodyti konstrukcinį karkasą	<i>View → Wireframe</i>		Trimačių objektų vaizdavimui naudoja „konstrukcinį karkasą“, neskaičiuoja šviesos šaltinių poveikio. „Konstrukcinis karkasas“ sudaromas, skaidant paviršius į trikampus (keturkampus) ir vaizduojant tik jų kraštines.
3.6	Rodyti paviršių	<i>View → Shading</i>		Trimačių objektų vaizdavimui naudoja paviršiaus normales ir skaičiuoja šviesos šaltinių poveikius.
3.7	Įjungti vieną švies. šaltinį	<i>View → 1 Light</i>		Objektų vaizdavimui naudoja vieną šviesos šaltinį.

3.8	Ijungti du švies. šalt.	<i>View</i> → <i>2 Lights</i>		Objektų vaizdavimui naudoja du šviesos šaltinius.
4.1	Versti spleiną	<i>2D Tools</i> → <i>Spline</i> <i>Conversion</i>		Verčia pasirinktas <i>NURBS</i> kreives į atkarpas arba atkarpas ir lankus, pagal pasirinktą žingsnį ir galimos kampo paklaidos reikšmę (žr.: 4.2.3 skyrelis).
4.2	Minimizuoti taškų kiekį	<i>2D Tools</i> → <i>Point Reduction</i>		Sumažina pasirinktų <i>NURBS</i> kreivių kontrolinių taškų skaičių, atsižvelgiant į užduotą atstumo, kampo paklaidą ir mažiausio atstumo tarp kontrolinių taškų reikšmę (žr.: 4.2.3 skyrelis).
4.3	Pakuoti dvimačius kontūrus	<i>2D Tools</i> → <i>Packing</i>		Išdėlioja dvimates figūras duotame plote, atsižvelgiant į užduotą minimalų atstumą tarp figūrų. Skaičiavimams naudojamas maišymo kriterijus ir užduotas skaičiavimui laikas. (žr.: 4.2.3 skyrelis).
5.1	Keisti srities proporcijas	<i>3D Tools</i> → <i>Region Scaling</i>		Keičia pasirinktų paviršių srities proporcijas išilgai vienos ar kelių ašių, atsižvelgiant į atsparos tašką. Taip pat galima pasirinkti sritis, kurios bus paslinktos, atsižvelgiant į pakeistas proporcijas (žr.: 4.2.3 skyrelis).
5.2	Skaičiuoti tūrį	<i>3D Tools</i> → <i>Volume</i> <i>Calculation</i>		Skaičiuoja erdvės tūrį, kurią apriboja pasirinkti paviršiai ir pagalbinės plokštumos. Skaičiavimai atliekami, atsižvelgiant į nurodytą segmentavimo žingsnio reikšmę (žr.: 4.2.3 skyrelis).
6.1	Nustatyti reikšmes	<i>Options</i> → <i>Default Values</i>		Nustatyto standartines atitinkamų parametrų reikšmes globaliai visiems algoritmams.
7.1	Rodyti/slėpti valdymo sektorių	<i>Window</i> → <i>Controls</i>		Paslepia kontrolės sektorių. Inicijavus šią funkciją dar kartą, rodo paslėptą sektorių.
7.2	Rodyti/slėpti informacijos sektorių	<i>Window</i> → <i>Information</i>		Paslepia informacijos sektorių. Inicijavus šią funkciją dar kartą – rodo paslėptą sektorių.
8.1	Informuoti apie produktą	<i>Help</i> → <i>About</i>		Atidaro informacijos apie produktą langą.

4.2.3. Funkcijų aprašymas



Splaininės kreivės vertimas į atkarpas ir lankus (4.1)



Step size: diskretizavimo žingsnis. Mažesnis žingsnis lemia didesnę tikslumą ir didesnę sugeneruotų esybių skaičių.

Angular tolerance: galima kampinė paklaida. Paklaida matuojama laipsniais, su mažesne reikšme skaičiavimai tikslesni, bet didesnis sugeneruotų atkarpų skaičius.

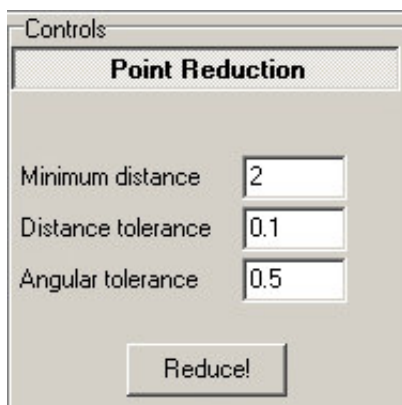
Convert to lines only: versti tik į atkarpas. Funkcija negeneruoja lankų.

Convert!: pradėti vertimą.

Inicijavus šią funkciją, sistema leidžia pasirinkti vieną ar kelias *NURBS* kreives, kurios bus verčiamos į atkarpas bei lankus. Jei pažymėtas „*versti tik į atkarpas*“ punktas, tada kreivės bus skaldomos į atkarpas (netikrinamos lankų sąlygos). Diskretizavimo žingsnis apytiksliai nurodo kokio ilgio atkarpomis skaidoma kreivė. Kolinearios atkarpos (sudarančios vieną tiesę) apjungiamos. Kolinearumo sąlyga tikrinama, apskaičiuojant kampą tarp besikertančių atkarpų. Jei atkarpos kertasi kampu lygiu $180^\circ \pm$ galima kampinė paklaida, jos traktuojamos kaip kolinearios. Funkcija negeneruoja $>180^\circ$ lankų, kad grubiai neiškraipyti kreivės.



Splaininės kreivės kontrolinių taškų kiekio minimizavimas (4.2)



Minimum distance: mažiausias atstumas tarp kontrolinių taškų. Atstumas tarp apdorotos kreivės kontrolinių taškų nebus mažesnis už šią reikšmę. Tai negalioja kampus formuojantiems taškams.

Distance tolerance: galima atstumo paklaida. Atstumas tarp kampus formuojančių taškų. Didesniam efektyvumui pasiekti reikia balansuojant šią reikšmę su minimalaus atstumo parametro reikšme.

Angular tolerance: galima kampinė paklaida. Paklaida matuojama laipsniais, su mažesne reikšme skaičiavimai tikslesni.

Reduce!: minimizuoti kontrolinių taškų skaičių.

Funkcija analizuoja pasirinktas *NURBS* kreives ir panaikina perteklinius kontrolinius taškus, jei

jų randa. Atsižvelgiant į splaininių kreivių specifiką, laikoma, kad keli taškai, esantys toje pačioje pozicijoje, naudojami aštraus kampo formavimui. Maksimalus kontrolinių taškų skaičius, reikalingas aštraus kampo formavimui priklauso nuo bazinės splaino funkcijos laipsnio (paprastai 3). Dėl įvairių netikslumų toleruojama kontrolinių taškų pozicijos paklaida, formuojant aštrų kampą. Ši paklaida įvertinama parametro „*galima atstumo paklaida*“ ir nurodo, koku atstumu esantys kontroliniai taškai, traktuojami kaip esantys toje pačioje pozicijoje. Tokių taškų pozicijos bus sutapatintos, jei jų yra daugiau nei reikia kampui formuoti, pertekliniai taškai bus pašalinti.

Galima kampinė paklaida įvertina kampo nuokrypį nuo 180° , skaičiuojant kampą tarp atkarpų, kurias formuoja trys kaimyniniai kontroliniai taškai. Kitaip, jei trys kontroliniai taškai yra vienoje tiesėje, tada vidurinis pašalinamas. Atsižvelgiant į mažiausio atstumo tarp kontrolinių taškų reikšmę, pašalinami taškai, kurie nutolę nuo kaimyninių taškų mažesniu atstumu. Tai negalioja taškams, kurie nutolę mažesniu atstumu, nei užduota galimos atstumo paklaidos.



Pakavimas dvimatėje erdvėje (4.3)

Controls	
Packing	
Packing area	3000 2000
Number of areas	1
Subsequence length	10
Tool diameter	5.0
Packing time	1 min.
Pack! Stop!	

Area: stačiakampio lapo, kuriame bus pakuojamos figūros, matmenys (išilgai $0x$ ir $0y$ ašių).

Number of areas: stačiakampių lapų, kuriuose galiam pakuoti, kiekis.

Subsequence length: sekos ilgis (maišymo kriterijus). Kuo didesnė reikšmė, tuo ilgesnis pilno skaičiavimo laikas, bet geresnis galimas dalinio skaičiavimo rezultatas.

Tool diameter: įrankio diametras.

Packing time: pakavimo laikas.

Pack!: pakuoti.

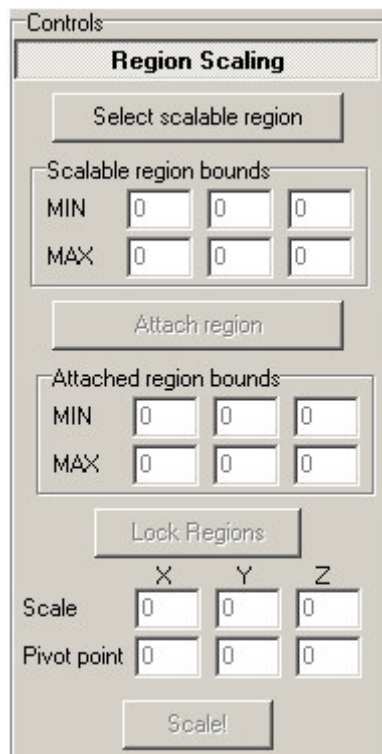
Stop!: sustabdyti skaičiavimus.

Funkcija leidžia pasirinkti figūras, kurias reikia išdėlioti stačiakampyje su užduotais matmenimis. Pasirinktų figūrų kontūrai negali kirstis. Kontūras kontūre traktuojamas kaip vidinė anga išorinio kontūro atžvilgiu. Figūros išdėliojamos įrankio diametro atstumu, stengiantis minimizuoti panaudotą tam plotą. Skaičiavimai vykdomi užduotą laiko tarpą (reikšmė minutėmis). Skaičiavimai gali būti priverstinai nutraukti anksčiau laiko, paspaudus „sustabdyti skaičiavimus“ mygtuką. Jei per užduotą laiką sistema suskaičiavo bent vieną pakavimo variantą, funkcija pateiks rezultatą. Jei buvo suskaičiuoti keli variantai – pateikiamas geriausias (su mažiausiomis ploto sąnaudomis). Sistema išveda rezultatus anksčiau nurodyto laiko, jei skaičiavimai užtruko mažiau.

Maišymo kriterijus įtakoja pozicionavimo lape sekos tvarką. Jei ši reikšmė lygi 1, tada figūros bus dëliojamos lape iš eilės, nuo didžiausios iki mažiausios. Jei ši reikšmė lygi figūrų skaičiui, tada visos sekos eiliškumas bus pakoreguotas atsitiktiniu būdu. Visos tarpinės reikšmės duos tarpinį rezultatą (pvz.: reikšmė 3 reiškia, kad pirma figūra gali būti sukeista vietomis su antra arba trečia, ketvirta su penkta arba šešta ir t.t.).



Modelio koregavimas nurodytoje srityje (5.1)



Select scalable region: pasirinkti koreguotiną sritį.

Scalable region bounds: koreguotinos srities ribos.

Attach region: susieti dar vieną sritį su koreguotina sritimi.

Attached region bounds: susietosios srities ribos.

Lock regions: fiksuoti sritis.

Scale (X Y Z): mastelio keitimo išilgai $0x$, $0y$ ir $0z$ ašių koeficientai.

Pivot point (X Y Z): atsparos taško koordinatės. Pagal nutylėjimą atsparos taškas yra geometrinis srities centras.

Scale: koreguoti (keisti proporcijas).

Paspaudus „*pasirinkti koreguotiną sritį*“ programa leidžia pasirinkti sritį iš vaizdavimo sektoriaus, naudojant pelę, kurios proporcijas reikia pakeisti. Srities ribas vëliau galima pakoreguoti, įrašant kitas reikšmes koreguotinos srities ribų laukuose. Taip pat gali būti pasirenkama susietoji sritis. Nustačius sričių ribas jas reikia užfiksuoti. Tada galiam nustatyti mastelio keitimo koeficientus, išilgai vienos ar kelių ašių (pvz.: atitinkamai 0,5; 2,0; 1,0 reikšmės reiškia, kad sritis bus dvigubai sumažinta išilgai $0x$ ašies ir dvigubai padidinta išilgai $0y$ ašies). Mastelio keitimo operacija vykdoma atsparos taško atžvilgiu.

Srities mastelio keitimas operuoja paviršių kontrolinių taškų pozicijomis. Pakoreguotos taškų pozicijos gali patekti į kaimyninių nekoreguojamų sričių erdvę. Daugeliu atveju tai yra nepageidautina, tokiu atveju reiktų naudoti susietąją sritį. Šios srities poabiai proporcijų nekeičia, bet yra paslenkami atitinkama kryptimi, kad koreguojamos srities kontroliniai taškai nepatektų į susietosios erdvę.



Tūrio skaičiavimas (5.2)

Controls

Volume Calculation

Step size

Boundary planes

	Min	Max
X	<input type="text" value="0"/>	<input type="text" value="0"/>
Y	<input type="text" value="0"/>	<input type="text" value="0"/>
Z	<input type="text" value="0"/>	<input type="text" value="0"/>

Step size: diskretizavimo žingsnis. Kuo mažesnis žingsnis, tuo didesnis tikslumas ir skaičiavimo laikas.

Boundary planes: pagalbinės ribojančios plokštumos.

Calculate!: skaičiuoti tūrį.

Funkcija leidžia pasirinkti paviršius, kurių ribojamos erdvės tūrį reikia skaičiuoti. Į skaičiavimus gali būti įtraukti apribojimai, apibrėžiami pagalbinių ribojančių plokštumų pagalba. Erdvė, kurios tūris skaičiuojamas, turi būti uždara. Skaičiavimai atliekami diskretizuojant paviršių pagal pasirinktą žingsnį.

4.2.4. Klaidų pranešimai

Funkcija	Klaida		Aprašymas
	English	Lietuviškai	
Volume Calculation	Error! Invalid Step size value	Klaida! Neteisinga žingsnio reikšmė	Žingsnis turi būti teigiamas.
	Error! Invalid boundary planes value	Klaida! Neteisingos ribojančių paviršių reikšmės.	Minimali reikšmė turi būti mažesnė už maksimalią.
Region Scaling	Error! Invalid Scalable region bounds value	Klaida! Neteisingos proporcijų keitimo ribos.	Minimali reikšmė turi būti mažesnė už maksimalią.
	Error! Invalid attached region bounds value	Klaida! Neteisingos gretimų paviršių ribos.	Minimali reikšmė turi būti mažesnė už maksimalią.
	Error! Invalid scale value.	Klaida! Neteisingos proporcingumo koeficientų	Proporcingumo koeficientų reikšmės turi

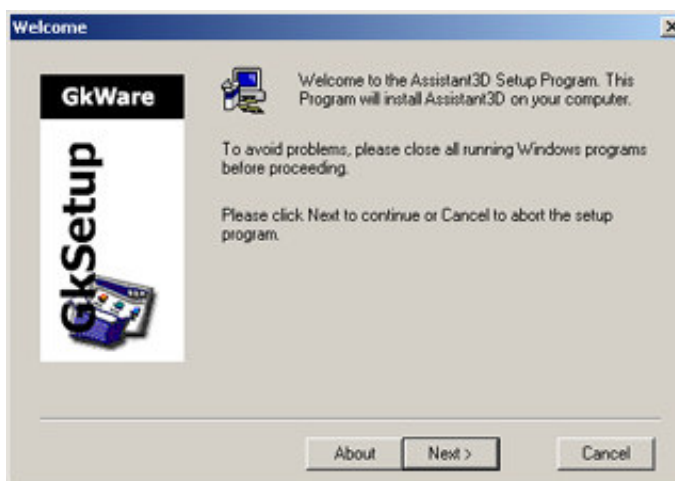
		reikšmės.	būti teigiamos
Spline Conversion	Error! Invalid Step size value	Klaida! Neteisinga žingsnio reikšmė.	Žingsnio reikšmė turi būti teigiama
	Error! Invalid Angular tolerance value	Klaida! Neteisinga kampų tolerancijos reikšmė.	Kampų tolerancijos reikšmė turi būti teigiama.
Point Reduction	Error! Invalid minimum distance value.	Klaida! Neteisinga minimalaus atstumo reikšmė.	Minimalaus atstumo reikšmė turi būti teigiama
	Error! Invalid Distance tolerance value	Klaida! Neteisinga atstumo tolerancijos reikšmė.	Atstumo tolerancijos reikšmė turi būti teigiama.
	Error! Invalid Angular tolerance value	Klaida! Neteisinga kampų tolerancijos reikšmė.	Kampų tolerancijos reikšmė turi būti teigiama.
Packing	Error! Invalid packing area value.	Klaida! Neteisinga pakavimo pločio ribos reikšmė.	Pakavimo pločio reikšmė turi būti teigiama.
	Error! Invalid number of areas.	Klaida! Neteisingas pakavimo plotų skaičius.	Pakavimo plotų skaičius turi būti teigiamas.
	Error! Invalid subsequence length value.	Klaida! Neteisingas poaibių skaičius.	Poaibių skaičius turi būti teigiamas.
	Warning! Subsequence length is greater then number of figures.	Dėmesio! Poaibių skaičius yra didesnis už figūrų skaičių.	Jeigu Poaibių skaičius didesnis tuomet seka nebus išskaidyta su poaibius, tokiu atveju pakavimo rezultatai gali būti prasti.
	Error! Invalid tool diameter value.	Klaida! Neteisingas pjūklo diametro skersmuo.	Pjūklo diametro reikšmė turi būti teigiama.

File	Error copying database template. Make sure igs_template.abs file exists and is not used by other programs	Klaida kopijuojant duomenų bazės šabloną. Įsitikinkite ar igs_template.abs byla egzistuoja ir nėra naudojama kitų programų.	Jei šios bylos nėra arba ją naudoja kitos programos, programa negalės dirbti.
	Error parsing IGES file. The file is corrupt.	Klaida nuskaityant duomenis iš IGES bylos. Byla yra išgadinta.	Esant klaidingiems duomenis byloje, programa negali atvaizduoti modelio bei atlikti redagavimo veiksmus.

Lentelė 4.2.1: Assistant3D klaidų pranešimai

4.3. Sistemos „Assistant3D“ įdiegimo vadovas

1. Pasileiskite bylą *SETUP.EXE* iš *Assistant3D* įdiegimo katalogo.
2. Atsidariusiame lange paspauskite mygtuką *Next* (žr.: Pav. 4.3.1).
3. Pasirinkite vietą diske kur bus įdiegta programa. Paspauskite mygtuką *Next*.
4. Sekančiame punkte parašykite pavadinimą meniu juostos kuri bus įdiegta į jūsų „*Start menu*“ juostą. Paspauskite *Next*.
5. Atsidariusiame lange patikrinkite ar visa informacija teisinga. Jeigu taip, paspauskite mygtuką *Next*, jeigu norite pakeisti informaciją paspauskite mygtuką *Back*.
6. Instaliacija baigta. Paspauskite mygtuką *Finish*.
7. Norėdami pasileisti programą, nueikite į katalogą į kurį įdiegėte programą ir pasileiskite bylą *Assistant3D.exe*.



Pav. 4.3.1: Assistant3D įdiegimo langas

5. IŠVADOS

Tyrimas parodė, jog šiuo metu sparčiai evoliucionuoja techninė įranga, skirta kompiuterizuotai gamybai. Dėl šios priežasties auga naujos programinės įrangos, atitinkančios naujus standartus poreikis. Taip pat pastebėta tendencija, jog augantis tokios programinės įrangos gamintojų skaičius, rinką daro nevientisa. Skirtingų kompanijų sukurtos *CAD/CAM* sistemos naudoja skirtingus duomenų formatus, todėl, siekiant realizuoti bendravimo galimybę, buvo sukurti keli duomenų mainų standartai.

IGES standarto tyrimas parodė, kad dauguma aprašytų elementų jau pasenę ir nenaudojami. Standartas perkrautas nereikalingu informacijos dvigubiniu. Didžiausias standarto trūkumas, jog skirtingos sistemos skirtingai interpretuoja tą pačią informaciją. Dėl šios priežasties labai reali svarbių duomenų praradimo tikimybė. Nepaisant to, didžioji rinkos dalis vis dar pasitiki *IGES* ir neskuba persiorientuoti. Populiariausia standarto dalis yra *NURBS* kreivės ir paviršiai, skirti geometrinių duomenų aprašymui.

NURBS analizės metu nustatyta, jog ši B-splaino modifikacija yra labai efektyvi. *NURBS* kreivė ar paviršius, racionalumo dėka, gali įgyti bet kokią formą, taip pat gali būti apskaičiuotas bet koku tikslumu. Kontrolinių taškų skaičius, reikalingas *NURBS* esybės aprašymui, yra santykinai mažas, todėl nenaudoja daug atminties resursų. *NURBS* buvo išbandytas su įvairiais algoritmais ir įrodė savo lankstumą.

Gaminio modelio analizavimo ir koregavimo algoritmų analizė parodė, jog egzistuoja nemažai dar nepublikuotų algoritmų modifikacijų. Dauguma jų buvo sėkmingai realizuoti, remiantis teoriniais išvedimais, ir davė neblogų rezultatų. Tačiau nagrinėta sritis nėra išsemta, galima padidinti algoritmų efektyvumą, išsamesnės analizės metu.

Šio darbo tikslas, ištirti gaminio kelią nuo modelio iki produkto, buvo sėkmingas. Išsamiau nenagrinėjant pačios gamybinės dalies, išanalizuoti visi modelio apdorojimo etapai, populiariausi naudojami įrankiai ir standartai. Išbandytos pasirinktos modelio analizavimo (tūrio skaičiavimo) ir koregavimo (srities proporcijų keitimo, *2D* pakavimo, *NURBS* kreivės apdorojimo) galimybės. Šio tyrimo pagrindu sukurta programinė įranga, įrodo praktinę darbo reikšmę ir analogiško projekto įgyvendinimo realumą. Visa šiame veikalė sukaupta informacija bus naudinga vėlesniems darbams, tiriantiems modelio duomenų apdorojimą, specializuotą *CAD/CAM* rinkai.

6. LITERATŪRA

- [1] ALBERT, M. *STEP NC—The End Of G-Codes? Iš MMS Online* [interaktyvus], 2006 [žiūrėta 2006-04-20]. Prieiga per Internetą:
<http://www.mmsonline.com/articles/070001.html>
- [2] BAKER, M. J. *Maths – Plane. Euclidean Space* [interaktyvus], 2005 [žiūrėta 2006-05-05]. Prieiga per Internetą:
<http://www.euclideanspace.com/maths/geometry/elements/plane/index.htm>
- [3] COURSE, D. *Developer roundtable: STEP vs. IGES Translations. Iš Mathdata* [interaktyvus], 2001 spalio [žiūrėta 2006-04-20]. Prieiga per Internetą:
http://www.mathdata.com/faqs/Developer-Roundtable_STEP-vs-IGES.php
- [4] DELTENER J. *3D Transformations. Iš Inverse Reality* [interaktyvus], [žiūrėta 2006-05-02]. Prieiga per Internetą:
<http://www.inversereality.org/tutorials/graphics%20programming/3dwmatrixes.html>
- [5] LAVOIE, P. *An introduction to NURBS*. 1999.
- [6] NELSON PUBLISHING, Inc. *CAD/CAM Software Report 2005. Iš Tooling & Production* [interaktyvus], 2006 [žiūrėta 2006-04-20]. Prieiga per Internetą:
http://www.manufacturingcenter.com/tooling/archives/1105/1105cad_cam.asp
- [7] NEWHOOK, P. *OpenGL and OS/2. All About NURBS. Iš EDM/2* [interaktyvus], 1998 [žiūrėta 2005-11-12]. Prieiga per Internetą:
<http://www.edm2.com/common/authors/pnewhook.html>
- [8] OWEN, K. *Area and Volume Calculations* [interaktyvus]. 2005 gruodis [žiūrėta 2006-01-15]. Prieiga per Internetą:
http://www.gamedev.net/community/forums/topic.asp?topic_id=279606
- [9] REID, E.; MOREA, G.; HARROD, D. A. Jr.; PELTZMAN, A.; PARKS, C.; GRUTTKE, W. B. *The Initial Graphics Exchange Specification (IGES) Version 6.0*, IGES/PDES Organization, 2001.
- [10] TRESELER, M.; FULLER, J.; LIMA, M.; SEAL, J.; DIXON, M. *OpenGL Programming Guide. Second Edition*. Addison-Wesley Publishing Company, 1997.

Model Analysis and Modification for Computer Aided Manufacturing

Summary

This research aims at the specifics of Computer Aided Design (*CAD*) and Computer Aided Manufacturing (*CAM*) market. Through discussion of the digital model's path since geometric construction till completion of a product the major trends are derived. Evolution of hardware for *CAM* challenges software developers to come up with new solutions. Growing numbers of *CAD/CAM* software proved to be considerable factor for incompatibility of data formats among systems. Therefore universal data exchange standards were claimed to enable communication. Also a few gaps of functionality among *CAD/CAM* software products encourages the development of new application, based on research. The set of functions to be integrated into application is supported theoretically according to researched information and tested practically during the realization of software.

The most successful standards and tools are selected as the basis of new software. *IGES* (*The Initial Graphics Exchange Specification*) standard along with *NURBS* (*Non-Uniform Rational B-Spline*) curves and surfaces is used for description of geometric data and *OpenGL* is used as drawing tool. Model analysis and modification for *CAM* includes following function description and realization: calculation of volume, non-uniform region scaling, *2D* packing, curve control point reduction, curve conversion to lines and arcs.

Testing results proves that most of the theoretic assumptions are correct and a development of such or similar software is truly possible. Nevertheless the effective improvement of discussed algorithms is possible through further research. All gathered information during this work will serve well for forthcoming researchers and software developers of modeling for *CAM*.

7. TERMINŲ IR SANTRUMPŲ ŽODYNAS

2D erdvė	Dvimatė erdvė, plokščia erdvė. Bet koks taškas dvimatėje erdvėje gali būti aprašytas dvejomis koordinatėmis – Ox ašyje ir Oy ašyje.
3D erdvė	Trimatė erdvė. Bet koks taškas dvimatėje erdvėje gali būti aprašytas trejomis koordinatėmis – Ox ašyje, Oy ašyje ir Oz ašyje.
4D erdvė	Keturmatė erdvė, homogeninė trimatė erdvė. Bet koks taškas keturmatėje erdvėje gali būti aprašytas dvejomis koordinatėmis – Ox ašyje, Oy ašyje, Oz ašyje ir svoriu w : (xw, yw, zw, w) , kur $w \neq 0$.
Apribotasis paviršius	Paviršius, kurio aprašas papildytas ribojančiomis esybėmis. Iškarpytojo paviršiaus analogas.
Antros eilės kreivė <i>ASCII</i>	Antros eilės kreivėms priklauso apskritimas, elipsė hiperbolė ir parabolė. <i>American Standard Code for Information Interchange</i> – plačiai paplitęs informacijos kodavimas, kurio pagrindas yra anglų kalbos abėcėle.
<i>Bezier</i> kreivė	Kreivė, kuri generuojama interpoliuojant kontrolinius taškus, naudojant n -tosios eilės polinomą.
B-splainas	Kreivė sudaryta iš <i>Bezier</i> segmentų, kurie susieti tam tikru tęstinumo laipsniu.
<i>CAD</i>	<i>Computer Aided Design</i> sistemos, specializuotos (gaminio) modelio projektavimui.
<i>CAM</i>	<i>Computer Aided Manufacturing</i> sistemos specializuotos, gaminio modelio apdirbimo programų sudarymui.
<i>DXF</i>	<i>Drawing Exchange Format</i> – formatas, skirtas duomenų tarp skirtingų sistemų pernešimui.
<i>IGES</i>	<i>Initial Graphics Exchange Specification</i> - formatas, skirtas duomenų tarp skirtingų sistemų pernešimui.
Iškarpančioji kreivė	Kreivė su kryptimi, apibrėžianti iškarpytojo paviršiaus ribas. Iškerpama paviršiaus dalis, esanti kreivės kairėje.
Iškarpytasis paviršius	Paviršius, kurio aprašas papildytas iškarpančiosiomis kreivėmis.
Kontrolinis taškas	Taškas, įtakojantis aibės tokių taškų aprašomo, objekto formą. <i>NURBS</i> kontrolinio taško poveikis yra lokalus ir priklauso nuo bazinės funkcijos laipsnio.

<i>NURBS</i>	Neunifikuotas racionalusis B-splainas – interpoliavimo metodas kreivėms ir paviršiams aprašyti, naudojamas kaip apibendrinimas <i>Bezier</i> kreivei. <i>NURBS</i> gali aprašyti antros eilės kreives.
Mazgų vektorius	Nemažėjančių realiųjų skaičių seka, naudojama <i>NURBS</i> kontrolinių taškų poveikiui lokalizuoti. Paprastai vektorių seka prasideda 0 ir baigiasi 1 reikšme.
Modelio erdvė	Erdvė, kurioje tiesiogiai išreikšta geometrinė modelio konstrukcija. Paprastai ją atitinka koordinačių sistema su pradžios tašku (0, 0, 0), kuriai galioja metrinės erdvės savybės ir kiekvienas taškas gali būti apibūdinamas trimis koordinatėmis (x, y, z). 2D modelio erdvei atitinkamai (x, y).
Parametrų erdvė	Parametrų (parametrinė) erdvė galioja tik vienos esybės ribose (kreivės, paviršiaus). Paprastai parametrų erdvę sudaro 2D (u, v) arba 1D (v) koordinačių sistema, kur koordinačių reikšmės kinta intervale [0; 1]. Esybės parametrų erdvė yra susijusi su modelio erdve per bazinę esybės funkciją. Kiekvienas (u, v) taškas gali būti išreikštas (x, y, z) koordinatėmis.
Ribojančioji esybė	Esybė, apibrėžianti paviršiaus ribas. Iškarpančiosios kreivės analogas.
Skleistinis paviršius	Paviršius, kuris generuojamas dviejų kreivių pagalba. Viena kreivė eian išilgai paviršiaus, kita – skersai.
Splainas	Speciali matematinė funkcija, aprašoma polinomais kiekvienam segmentui. Dažnai naudojamas interpoliuojant.
<i>STEP</i>	<i>Standard for the Exchange of Product</i> - formatas, skirtas duomenų tarp skirtingų sistemų pernešimui.

8. PRIEDAI

8.1. IGES formato esybių tipų aprašymas

Kiekvienam esybės tipui parametrų kiekis, išdėstymas ir prasmė skiriasi. Bendras bruožas visiems tipam yra pirmasis įrašo parametras (esybių tipų aprašyme jis bus laikomas 0-uoju parametru ir bus praleidžiamas) parametrų sekcijoje. Jis nurodo esybės tipo numerį (žr. *Lentelė 8.1.1*).

Esybės tipo numeris	Esybės tipas
100	Apskritas lankas (<i>Circular Arc</i>)
102	Sudėtinė kreivė (<i>Composite Curve</i>)
104	Lankas antros eilės kreivė pagrindu (<i>Conical Arc</i>)
106	Duomenų masyvas (<i>Copious Data</i>)
106/11	2D laužtė (<i>2D Line Path</i>)
106/12	3D laužtė (<i>3D Line Path</i>)
106/63	Paprastoji uždaroji plokščioji kreivė (<i>Simple Closed Planar Curve</i>)
108	Plokštuma (<i>Plane</i>)
110	Atkarpa (<i>Line</i>)
112	Parametrinė splaininė kreivė (<i>Parametric Spline Curve</i>)
114	Parametrinis splaininis paviršius (<i>Parametric Spline Surface</i>)
116	Taškas (<i>Point</i>)
118	„Ruled“ paviršius (<i>Ruled Surface</i>)
120	Simetrinis ašiai paviršius (<i>Revolution Surface</i>)
122	Skleistiniai paviršiai (<i>Tabulated Cylinder</i>)
124	Transformacijų matrica (<i>Transformation Matrix</i>)
125	Pagalbinės esybės (<i>Flash</i>)
126	Racionalioji B-splaininė kreivė (<i>Rational B-Spline Curve</i>)
128	Racionalusis B-splaininis paviršius (<i>Rational B-Spline Surface</i>)
130	Paslinktoji kreivė (<i>Offset Curve</i>)
140	Paslinktasis paviršius (<i>Offset Surface</i>)
141	Ribojančioji esybė (<i>Boundary</i>)
142	Kreivė parametriniame paviršiuje (<i>Curve on Parametric Surface</i>)
143	Apribotasis paviršius (<i>Bounded Surface</i>)
144	Iškarpytasis parametrinis paviršius (<i>Trimmed Parametric Surface</i>)
314	Spalvos esybė (<i>Color Definition Entity</i>)

Lentelė 8.1.1: IGES byloje naudojamų geometrinių esybių tipų numerių reikšmės, žaliai pažymėti nagrinėtini tipai

Parametrai vienas nuo kito atskirti skirtuku (žr.: *Lentelė 2.6.7*), jų reikšmės gali priklausyti sveikųjų arba realiųjų skaičių aibėms. Sveikaisiais skaičiais aprašomos rodyklės, tipų vėliavėlės, realiaisiais – koordinatės, svoriai ir pan.

Geometriniam duomenim aprašyti naudojama trimatė Euklido erdvė. Kiekviena esybė gali turėti atskirą koordinačių pradžios tašką, taip pat individualią koordinačių sistemos orientaciją (toku atveju esybės erdvės koordinatės žymimos taip: X_T, Y_T, Z_T ; modelio erdvės koordinatės: X, Y, Z). Tada esybės koordinatės perskaičiuojamos į modelio erdvę (su koordinačių sistema bendra visam modeliui), naudojant konkrečią transformacijų matricą.

8.1.1. Apskritas lankas (100)

Aprašymas: Lankas, kurio bazinė funkcija yra apskritimas, kurio lygtis:

$$R = \sqrt{(X_i - X_0)^2 + (Y_i - Y_0)^2},$$

kur R – apskritimo spindulys, X_0, Y_0 – apskritimo centras, $i = 1, 2$; Lankas apibrėžtas taškuose:

$$(R \cos(t_i), R \sin(t_i)) = (X_i - X_0, Y_i - Y_0), 0 \leq t_1 \leq 2\pi, 0 \leq t_2 - t_1 \leq 2\pi,$$

kur t_i – lanko pradžios ir pabaigos kampai, X_i, Y_i – lanko pradžios ir pabaigos taškai.

Esybės koordinačių sistema parinkta taip, kad lankas būtų $X_T Y_T$ plokštumoje arba plokštumoje lygiagrečioje $X_T Y_T$.

D sekcijos parametrai: Parametrų reikšmių tipas ir diapazonas apibrėžtas (žr.: *Lentelė 8.1.2*).

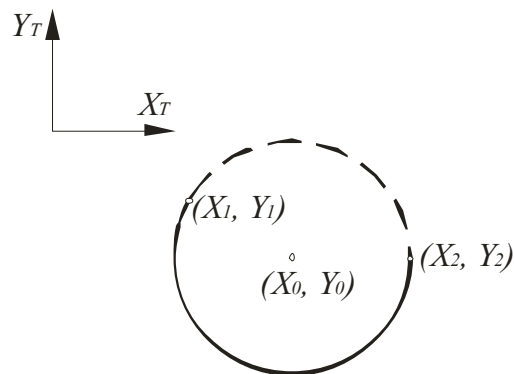
Nr.	Parametras	Reikšmės
1	Esybės tipo numeris	100
7	Transformacijų matrica	0, \Rightarrow
9a	Būsena: matomumas	00 – 01
9b	Būsena: priklausomybė	00 – 03
9c	Būsena: paskirtis	00
9d	Būsena: hierarchija	Ignoruojama
13	Spalvos numeris	#, \Rightarrow
15	Forma	0

Lentelė 8.1.2: Apskrito lanko D sekcijos parametrai

P sekcijos parametrai:

<u>Nr.</u>	<u>Vardas</u>	<u>Tipas</u>	<u>Aprašymas</u>
1	Z_T	Realusis	Poslinkis Z_T ašies kryptimi.
2	X_1	Realusis	Lanko centro abscisė.
3	Y_1	Realusis	Lanko centro ordinatė.
4	X_2	Realusis	Pradžios taško abscisė.
5	Y_2	Realusis	Pradžios taško ordinatė.
6	X_3	Realusis	Pabaigos taško abscisė.
7	Y_3	Realusis	Pabaigos taško ordinatė.

Pavyzdžiai:



Pav. 8.1.1: Apskritas lankas

8.1.2. Atkarpa (110)

Aprašymas: Atkarpa yra tiesė apribota iš abiejų galų pradžios (P_1) ir pabaigos (P_2) taškais. Kiekvienas iš taškų apibrėžtas trimis modelio erdvės koordinatėmis, jų pakanka nedviprasmiškai apibrėžti atkarpa, jei tik $P_1 \neq P_2$.

D sekcijos parametrai: Parametrų reikšmių tipas ir diapazonas apibrėžtas (žr.: *Lentelė 8.1.3*).

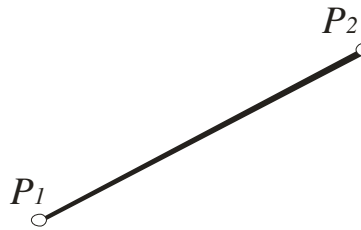
Nr.	Parametras	Reikšmės
1	Esybės tipo numeris	110
7	Transformacijų matrica	0, \Rightarrow
9a	Būsena: matomumas	00 – 01
9b	Būsena: priklausomybė	00 – 03
9c	Būsena: paskirtis	00
9d	Būsena: hierarchija	Ignoruojama
13	Spalvos numeris	#, \Rightarrow
14	Forma	0, (1, 2)

Lentelė 8.1.3: Atkarpos D sekcijos parametrai

P sekcijos parametrai:

<u>Nr.</u>	<u>Vardas</u>	<u>Tipas</u>	<u>Aprašymas</u>
1	X_1	Realusis	Pradžios taško P_1 koordinatė OX ašyje.
2	Y_1	Realusis	Pradžios taško P_1 koordinatė OY ašyje.
3	Z_1	Realusis	Pradžios taško P_1 koordinatė OZ ašyje.
4	X_2	Realusis	Pabaigos taško P_2 koordinatė OX ašyje.
5	Y_2	Realusis	Pabaigos taško P_2 koordinatė OY ašyje.
6	Z_2	Realusis	Pabaigos taško P_2 koordinatė OZ ašyje.

Pavyzdžiai:



Pav. 8.1.2: Atkarpa

8.1.3. Transformacijų matrica (124)

Aprašymas: Transformacijų matricos paskirtis yra atvaizduoti vienos koordinačių sistemos taškus kitoje koordinačių sistemoje. Transformavimas atliekamas veiksmų su matricomis pagalba:

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} X_l \\ Y_l \\ Z_l \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} X_{lš} \\ Y_{lš} \\ Z_{lš} \end{bmatrix}.$$

Čia vektorius $[X_l \ Y_l \ Z_l]$ yra taškas koordinačių sistemoje prieš transformaciją, $[X_{lš} \ Y_{lš} \ Z_{lš}]$ – po transformacijos. $R = [R_{ij}]$ – realiųjų skaičių matrica, $T = [T_i]$ – realiųjų skaičių vektorius, $i, j = 1, 2, 3$. Transformacijų matricos esybę aprašoma šiais 12 parametrų ($i * j + i$) parametrų sekcijoje.

Jei koordinatės modelio erdvėje gaunamos transformavus esybės erdvės taškus naudojant R_1 ir T_1 , o po to dar karta transformavus naudojant R_2 ir T_2 , tada galima apskaičiuoti tiesioginę transformaciją pagal:

$$R = R_1 \times R_2, \quad T = R_2 \times T_1 + T_2.$$

Yra kelios transformacijų matricos esybių formų, kurios leidžia apibrėžti esybės pobūdį. Formos numeris gali būti nuskaitytas iš D sekcijos 14 parametro. Reikšmė „0“ (standartinė) reiškia, kad R yra orto-normuota su determinantu lygiu 1. Reikšmė „1“ reiškia, kad R yra orto-normuota su determinantu lygiu -1. Reikšmės „10“, „11“, „12“ šiame darbe neaktualios.

D sekcijos parametrai: Parametrų reikšmių tipas ir diapazonas apibrėžtas (žr.: Lentelė 8.1.4).

Nr.	Parametras	Reikšmės
1	Esybės tipo numeris	124
7	Transformacijų matrica	0, \Rightarrow
9a	Būsena: matomumas	Ignoruojama
9b	Būsena: priklausomybė	Ignoruojama
9c	Būsena: paskirtis	00
9d	Būsena: hierarchija	Ignoruojama
13	Spalvos numeris	#, \Rightarrow
14	Forma	0,1, (10, 11, 12)

Lentelė 8.1.4: Transformacijų matricos D sekcijos parametrai

P sekcijos parametrai:

<u>Nr.</u>	<u>Vardas</u>	<u>Tipas</u>	<u>Aprašymas</u>
1	R_{11}	Realusis	Viršutinė R matricos eilutė.
2	R_{12}	Realusis	...
3	R_{13}	Realusis	...
4	T_1	Realusis	Viršutinis T vektoriaus narys.
5	R_{21}	Realusis	Vidurinė R matricos eilutė.
6	R_{22}	Realusis	...
7	R_{23}	Realusis	...
8	T_2	Realusis	Vidurinė T vektoriaus narys.
9	R_{31}	Realusis	Apatinė R matricos eilutė.
10	R_{32}	Realusis	...
11	R_{33}	Realusis	...
12	T_3	Realusis	Apatinis T vektoriaus narys.

8.1.4. Racionalioji B-splaininė kreivė (126)

Aprašymas: Racionalusis B-splainas naudojamas kreivėm braižyti. Interpoliuojant tarp nedidelio skaičiaus kontrolinių taškų galima išgauti gan sudėtingas ir tikslias kreives. B-splainas yra *Bezier* kreivės apibendrinimas, realizuojamas naudojant papildomą mazgų vektorių (*knot vector*). Pastarasis suteikia B-splainui lokalizavimo savybę, taip išvengiant vieno kontrolinio taško poveikio visam splainui. Detalus *NURBS* aprašymas 2.5 skyriuje.

IGES standartas turi kelias dokumentuotas racionaliojo B-splaino formas (formos numeris patalpintas *D* sekcijos 14 parametro lauke žr.: *Lentelė 8.1.5*):

- Formos numeris „0“ – apskaičiuojama iš B-splaino parametrų.
- Formos numeris „1“ – atkarpa.
- Formos numeris „2“ – apskritas lankas.
- Formos numeris „3“ – elipsinis lankas.
- Formos numeris „4“ – parabolinis lankas.
- Formos numeris „5“ – hiperbolinis lankas.

Šiame darbe analizuojamas B-splaino interpretacija tik su „0“ formos numeriu.

Jei kreivė apibrėžta konkrečioje plokštumoje, tada plokštumo vėliavėlei (*PROP1*) nustatoma reikšmė „1“, kitaip „0“. Jei *PROP1 = 1*, tada iš paskutinių trijų įrašo P sekcijoje parametrų galima nuskaityti plokštumos, kurioje yra splainas normalės vektorius.

Jei kreivės galų taškai sutampa, tada kreivė uždara ir *PROP2 = 1*, priešingu atveju *PROP2 = 0*.

Jei kreivė racionalusis B-splainas ir visi kontrolinių taškų svoriai nėra lygūs, tada *PROP3 = 0*, kitaip *PROP3 = 1* – kreivė polinominė.

Jei kreivė periodinė - *PROP3 = 1*, kitaip *PROP3 = 0*.

Pastebėtini dėsniumai:

- *CPN* – kontrolinių taškų skaičius (parastai taškų skaičius generuojančioje programoje + 2, kai du pirmi taškai sutampa, du paskutiniai taip pat).
- *M* – bazinės funkcijos laipsnis (paprastai *M = 3*).
- *K = CPN - 1* – kontrolinių taškų skaičius, indeksuojant nuo 0.
- *KN = K + M + 2* – elementų kiekis mazgų vektoriuje.

D sekcijos parametrai: Parametrų reikšmių tipas ir diapazonas apibrėžtas (žr.: *Lentelė 8.1.5*).

Nr.	Parametras	Reikšmės
1	Esybės tipo numeris	126
7	Transformacijų matrica	0, ⇒
9a	Būsena: matomumas	00 – 01
9b	Būsena: priklausomybė	00 – 03
9c	Būsena: paskirtis	00, 05
9d	Būsena: hierarchija	Ignoruojama
13	Spalvos numeris	#, ⇒
14	Forma	0, (1 - 5)

Lentelė 8.1.5: Racionaliojo B-splaino kreivės D sekcijos parametrai

P sekcijos parametrai:

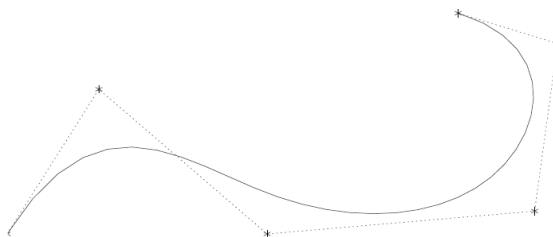
<u>Nr.</u>	<u>Vardas</u>	<u>Tipas</u>	<u>Aprašymas</u>
1	<i>K</i>	Sveikasis	Kontrolinių taškų skaičius, indeksuojant nuo 0.
2	<i>M</i>	Sveikasis	Bazinės funkcijos laipsnis.
3	<i>PROP1</i>	Sveikasis	Plokštumo vėliavėlė.
4	<i>PROP2</i>	Sveikasis	Uždarumo vėliavėlė.
5	<i>PROP3</i>	Sveikasis	Racionalumo vėliavėlė.
6	<i>PROP4</i>	Sveikasis	Periodiškumo vėliavėlė.

Tegul $A = K + M + 1$, $N = K - M + 1$

7	$T(-M)$	Realusis	Pirmoji mazgų vektoriaus reikšmė.
	...		
$7+A$	$T(N+M)$	Realusis	Paskutinioji mazgų vektoriaus reikšmė.
$8+A$	$W(0)$	Realusis	Pirmojo kontrolinio taško svoris.
	...		
$8+A+K$	$W(K)$	Realusis	Paskutiniojo kontrolinio taško svoris.
$9+A+K$	$X(0)$	Realusis	Pirmojo kontrolinio taško koordinatės.
$10+A+K$	$Y(0)$	Realusis	...
$11+A+K$	$Z(0)$	Realusis	...
	...		
$9+A+4K$	$X(K)$	Realusis	Paskutiniojo kontrolinio taško koordinatės.
$10+A+4K$	$Y(K)$	Realusis	...
$11+A+4K$	$Z(K)$	Realusis	...
$12+A+4K$	$V(0)$	Realusis	Mažiausia (pirmoji) mazgų vektoriaus reikšmė.
$13+A+4K$	$V(1)$	Realusis	Didžiausia (paskutinioji) mazgų vektoriaus reikšmė.
$14+A+4K$	$XNORM$	Realusis	Plokštumos normalė (jei $PROPI = 1$).
$15+A+4K$	$YNORM$	Realusis	...
$15+A+4K$	$ZNORM$	Realusis	...

Pastebėjimai: Pagal apibrėžimą mazgų vektoriaus elementų reikšmės turi būti diapazone $[0, 1]$, kaip matyti iš pavyzdžio (žr.: *Lentelė 2.6.1, eilutė 135*) maksimali riba viršija 1. Todėl reikia normalizuoti mazgų vektoriaus elementų reikšmes.

Pavyzdžiai:



Pav. 8.1.3: Racionalioji B-splaininė kreivė

8.1.5. Racionalusis B-splaininis paviršius (128)

Aprašymas: Racionalusis B-splaino paviršius generuojamas interpoliuojant keletą B-splaino kreivių. Apskaičiuot visą splaino kreivės reikšmes reikia vieno parametro, o paviršiui reikia dviejų: vienas kreivei išilgai paviršiaus (paviršiams paprastai vadinamas u parametriniu kintamuoju), kitas

– skersai paviršiaus (vadinamas v parametriniu kintamuoju). Detalus *NURBS* aprašymas 2.5 skyriuje.

Jei pirmoji generuojanti kreivė yra uždara – $PROP1 = 1$, kitaip $PROP1 = 0$. Jei antroji generuojanti kreivė uždara – $PROP2 = 1$, kitaip $PROP2 = 0$.

Jei paviršius racionalusis (žr.: 8.1.4 skyrelyje) – $PROP3 = 0$, kitaip $PROP3 = 1$.

Jei paviršius periodinis pagal pirmąją parametrinę kintamąją – $PROP4 = 1$, kitaip $PROP4 = 0$. Jei paviršius periodinis pagal antrąją parametrinę kintamąją – $PROP5 = 1$, kitaip $PROP5 = 0$.

Paviršiui gali būti priskirtas formos numeris nuo „0“ iki „9“. Šiame darbe analizuojama tik racionaliojo B-splaininio paviršiaus interpretacija su formos numeriu „0“, kada forma iš anksto nenustatyta ir paviršiaus generavimas apibrėžiamas parametru reikšmėmis iš *P* sekcijos. Kitais atvejais:

- Formos numeris „1“ – plokštuma.
- Formos numeris „2“ – apskritas cilindras.
- Formos numeris „3“ – kūgis.
- Formos numeris „4“ – sfera.
- Formos numeris „5“ – žiedas (simetrinis ašiai kūnas, kurio generuojanti kreivė yra apskritimas) (*torus*).
- Formos numeris „6“ – atkarpa.
- Formos numeris „7“ – skleistinis paviršius.
- Formos numeris „8“ – „Ruled“ paviršius.
- Formos numeris „9“ – „Quadric“ paviršius.

D sekcijos parametrai: Parametru reikšmių tipas ir diapazonas apibrėžtas (žr.: *Lentelė 8.1.6*).

Nr.	Parametras	Reikšmės
1	Esybės tipo numeris	128
7	Transformacijų matrica	0, \Rightarrow
9a	Būsena: matomumas	00 – 01
9b	Būsena: priklausomybė	00 – 03
9c	Būsena: paskirtis	00
9d	Būsena: hierarchija	Ignoruojama
13	Spalvos numeris	#, \Rightarrow
14	Forma	0, (1 - 9)

Lentelė 8.1.6: Racionaliojo B-splaininio paviršiaus D sekcijos parametrai

P sekcijos parametrai:

<u>Nr.</u>	<u>Vardas</u>	<u>Tipas</u>	<u>Aprašymas</u>
1	$K1$	Sveikasis	Kontrolinių taškų skaičius pirmajai generuojančiai kreivei, indeksuojant nuo 0.
2	$K2$	Sveikasis	Kontrolinių taškų skaičius antrajai generuojančiai kreivei, indeksuojant nuo 0.
3	$M1$	Sveikasis	Pirmojo splaino bazinės funkcijos laipsnis.
4	$M2$	Sveikasis	Antrojo splaino bazinės funkcijos laipsnis.
5	$PROP1$	Sveikasis	Uždarumo vėliavėlė pirmajai generuojančiai kreivei.
6	$PROP2$	Sveikasis	Uždarumo vėliavėlė antrajai generuojančiai kreivei.
7	$PROP3$	Sveikasis	Racionalumo vėliavėlė.
8	$PROP4$	Sveikasis	Periodiškumo vėliavėlė pirmajai generuojančiai kreivei.
9	$PROP5$	Sveikasis	Periodiškumo vėliavėlė antrajai generuojančiai kreivei.
<p>$Tegul A = K1 + M1 + 1, N1 = K1 - M1 + 1, B = K2 + M2 + 1, N2 = K2 - M2 + 1,$ $C = (K1+1)*(K2+1)$</p>			
10	$T(-M1)$	Realusis	Pirmojo mazgų vektoriaus pirmoji reikšmė.
	...		
10+A	$T(N1+M1)$	Realusis	Pirmojo mazgų vektoriaus paskutinioji reikšmė.
11+A	$T(-M2)$	Realusis	Antrojo mazgų vektoriaus pirmoji reikšmė.
	...		
11+A+B	$T(N2+M2)$		Antrojo mazgų vektoriaus paskutinioji reikšmė.
12+A+B	$W(0,0)$	Realusis	Pirmojo kontrolinio taško svoris.
13+A+B	$W(1,0)$	Realusis	Antrojo kontrolinio taško svoris.
	...		
11+A+B+C	$W(K1,K2)$	Realusis	Paskutiniojo kontrolinio taško svoris.
12+A+B+C	$X(0,0)$	Realusis	Pirmojo kontrolinio taško koordinatės.
13+A+B+C	$Y(0,0)$	Realusis	...
14+A+B+C	$Z(0,0)$	Realusis	...
15+A+B+C	$X(1,0)$	Realusis	Antrojo kontrolinio taško koordinatės.
16+A+B+C	$Y(1,0)$	Realusis	...
17+A+B+C	$Z(1,0)$	Realusis	...

	...		
$9+A+B+4C$	$X(K1,K2)$	Realusis	Paskutiniojo kontrolinio taško koordinatės.
$10+A+B+4C$	$Y(K1,K2)$	Realusis	...
$11+A+B+4C$	$Z(K1,K2)$	Realusis	...
$12+A+B+4C$	$U(0)$	Realusis	Mažiausia pirmojo mazgų vektoriaus reikšmė.
$13+A+B+4C$	$U(1)$	Realusis	Didžiausia pirmojo mazgų vektoriaus reikšmė.
$14+A+B+4C$	$V(0)$	Realusis	Mažiausia antrojo mazgų vektoriaus reikšmė.
$15+A+B+4C$	$V(1)$	Realusis	Didžiausia antrojo mazgų vektoriaus reikšmė.

8.1.6. Ribojančioji esybė (141)

Aprašymas: Kiekviena ribojančioji esybė nurodo paviršiaus ribas, aprašomas kreivių, einančių paviršiumi. Paviršiumi, jo riboms ir ribas aprašančioms kreivėms būdingos savybės:

D1. $S(u, v)$ gali būti naudojamas kaip parametrinis paviršius su ribojančia esybe, jei atitinka tokius reikalavimus:

- Neiškarpyta paviršiaus $S(u, v)$ sritis D yra stačiakampis, sudarytas iš (u, v) taškų, tenkinančių nelygybes $a \leq u \leq b$, $c \leq v \leq d$ su duotom konstantom $a < b$ ir $c < d$.
- Projekcija $S = S(u, v) = (x(u, v), y(u, v), z(u, v))$ apibrėžta kiekvienai (u, v) porai srityje D .
- Projekcija yra vienas prie vieno srities D viduje (bet nebūtinai jos riboje).
 - Projekcija turi normalės vektorius kiekvienam srities D taške, išskyrus taškus suprojektuotus D į polius (žr.: savybė D3).

D2. Izo-parametrinės kreivės $u = a$, $u = b$, $v = c$, $v = d$ vadinamos ribojančiomis kreivėmis parametrinėje erdvėje arba tiesiog ribojančiomis kreivėmis.

D3. Tarkim, kad P yra 3D taškas modelio erdvėje. P yra paviršiaus polius, apibrėžtas projekcijos $S(u, v)$, jei tenkina bent viena iš sąlygų:

- $P = S(a, v)$ su visais v , kur $c \leq v \leq d$.
- $P = S(b, v)$ su visais v , kur $c \leq v \leq d$.
- $P = S(u, c)$ su visais u , kur $a \leq u \leq b$.
- $P = S(u, d)$ su visais u , kur $a \leq u \leq b$.

D4. Tarkim, kad C yra 3D kreivė modelio erdvėje. C yra paviršiaus siūlė, apibrėžta projekcijos $S(u, v)$, jei ji yra atvaizdas modelio erdvėje:

- $C(v) = S(a, v)$ su visais v , kur $c \leq v \leq d$ ir

$C(v) = S(b, v)$ su visais v , kur $c \leq v \leq d$.

arba

b) $C(u) = S(u, c)$ su visais u , kur $a \leq u \leq b$ ir

$C(u) = S(u, d)$ su visais u , kur $a \leq u \leq b$.

D5. Modelio erdvės kreivė išreikšta parametriškai ir savęs nekerta nė vienam taške, tačiau kreivės galai gali būti sujungti (uždara).

D6. Riba(-os) yra modelio erdvės kreivių seka $(C_i, i = 1, n)$ su šiomis savybėmis:

a) Seka uždara: C_n pabaigos taškas yra C_1 pradžios taškas.

b) Kiekviena kreivė sekoje orientuota taip, kad C_{i-1} pabaigos taškas sutaptų su C_i pradžios tašku, kur $i = 2, n$.

c) Sekos elementai nesikerta tarpusavyje, išskyrus C_n pabaigos tašką ir C_1 pradžios tašką. Sekos elementai nekerta kitų ribų (kreivių sekų), nebent izoliuotuose taškuose (žr.: savybė D10(b)).

D7. Modelio erdvės iškarpančioji kreivė naudojama, atsižvelgiant į orientaciją. Ji yra sekos, formuojančios ribą, dalis.

D8. Teigiama paviršiaus normalė gaunama vektoriškai sudauginus $\partial S(u, v) / \partial u$ ir $\partial S(u, v) / \partial v$.

D9. Terminologija „į kairę nuo modelio erdvės iškarpančiosios kreivės taške p “ reiškia „taške p , esančio paviršiaus normalės ir iškarpančiosios kreivės liestinės vektorinės sandaugos kryptimi“.

D10. Paviršiaus regionas, kuris yra analizuojamas, vadinamas aktyviuoju. Jis tenkina tokias sąlygas:

a) Aktyvusis regionas turi baigtinį plotą.

b) Bet kurie du aktyviojo regiono vidaus taškai yra sujungti keliu.

c) Aktyvusis regionas apibrėžiamas ribomis, jo vidumi vadinama paviršiaus dalis, esanti ribojančioms kreivėms iš kairės.

d) Aktyviajam regionui priklauso jo vidus ir ribos.

e) Aktyviojo regiono vidaus uždara sritis yra aktyvusis regionas.

D11. Sakykim, kad C_a^* yra susietoji parametru erdvės kreivė lanko C_a iš modelio erdvės iškarpančiosios kreivės C ant paviršiaus S su sritimi D . Jei C_a^* priklauso D , tada kompozicija $S \circ C_a^* = C_a$. Susietoji parametru erdvės kreivė išreiškiami parametriškai ir nekerta savęs nė vienam taške, tačiau kreivės galai gali būti sujungti.

D12. Susietųjų parametrų erdvės kreivių kolekcija (arba tiesiog „kolekcija“) susideda iš tokių susietųjų parametrų erdvės kreivių (C_i^* , $i = 1, p$), kad C_i gaunamos iš sudėtinės kreivės kompozicijos ($S \circ C_i^*$, $i = 1, p$). Sudėtinė kreivė sudaryta iš sekos su C_i elementais, orientuotais taip, kad parametrai kintant nuo jo pradinės iki galinės reikšmės, sugeneruojama pilna modelio erdvės iškarpančioji kreivė su kryptimi, nurodyta modelio erdvės kreivės vėliavėlėje *SENSE*.

Susietosios parametrų erdvės kolekcijos, sudarytos iš C_i^* , riboms ne būtinai turi tenkinti uždarumo sąlyga (žr.: sąlyga D6). C_i^* gali formuoti ribas, pridėdant atitinkamas sekcijas ribojančių kreivių parametrų erdvėje (žr.: sąlyga D2).

D sekcijos parametrai: Parametrų reikšmių tipas ir diapazonas apibrėžtas (žr.: Lentelė 8.1.7).

Nr.	Parametras	Reikšmės
1	Esybės tipo numeris	141
7	Transformacijų matrica	0, \Rightarrow
9a	Būsena: matomumas	00 – 01
9b	Būsena: priklausomybė	00 – 03
9c	Būsena: paskirtis	00
9d	Būsena: hierarchija	Ignoruojuama
13	Spalvos numeris	#, \Rightarrow
14	Forma	0

Lentelė 8.1.7: Ribojančios esybės D sekcijos parametrai

P sekcijos parametrai:

<u>Nr.</u>	<u>Vardas</u>	<u>Tipas</u>	<u>Aprašymas</u>
1	<i>TYPE</i>	Sveikasis	Apribotojo paviršiaus išraiškos tipas: „0“ – ribojančios esybės turės nuorodas tik į modelio erdvės iškarpančiąsias kreives. Susietoji paviršiaus išraiška (rodyklė iš <i>STPR</i>) gali būti parametrinė. „1“ – ribojančios esybės turės nuorodas į modelio erdvės kreives bei į susietąsias parametrų erdvės kolekcijas. Susietasis paviršius (rodyklė iš <i>STPR</i>) parametrinės išraiškos.
2	<i>PREF</i>	Sveikasis	Iškarpančiųjų kreivių išraiškos prioritetas siūnciančioje sistemoje: „0“ – neapibrėžta.

			„1“ – modelio erdvės.
			„2“ – parametrų erdvės.
			„3“ – vienodo prioriteto.
3	<i>SPTR</i>	Rodyklė	Rodyklė į esybę, aprašančią neiškarpytą paviršių, kurį reiki apriboti, <i>D</i> sekcijoje. Jei <i>TYPE = 1</i> , tada paviršiaus išraiška bus parametrinė.
4	<i>N</i>	Sveikasis	Kreivių, naudojamų šioje ribojančioje esybėje kiekis.
5	<i>CRVPNT(1)</i>	Rodyklė	Rodyklė į pirmąją modelio erdvės kreivę (<i>D</i> sekcijoje), naudojamą šioje ribojančioje esybėje.
6	<i>SENSE(1)</i>	Sveikasis	Orientacijos vėliavėlė, nurodanti ar pirmosios modelio erdvės kreivės kryptis turėtų būti apversta prieš formuojant ribą. Galimos reikšmės: „1“ – <i>PSCPT</i> ir <i>CRVPT</i> kryptys sutampa, apversti nereikia. „2“ - <i>PSCPT</i> ir <i>CRVPT</i> kryptys yra priešingos – reikia apversti.
7	<i>K(1)</i>	Sveikasis	Asocijuotųjų parametrų erdvės kreivių kiekis, panaudotas sudaryti kolekcijai, formuojančiai pirmąją modelio erdvės iškarpančiąją kreivę. Jei <i>TYPE = 0</i> , tada ši reikšmė bus „0“.
8	<i>PSCPT(1,1)</i>	Rodyklė	Rodyklė į esybę <i>D</i> sekcijoje, kuri yra pirmoji susietoji parametrų erdvės kreivė, naudojama kolekcijai, formuojančiai pirmąją modelio erdvės iškarpančiąją kreivę.
<i>Tegul $M = 12 + 3*(N-1) + (K(1) + K(2) + \dots + K(N-1))$</i>			
$7+K(1)$	<i>PSCPT(1,K(1))</i>	Rodyklė	Rodyklė į esybę <i>D</i> sekcijoje, kuri yra paskutinioji susietoji parametrų erdvės kreivė, naudojama kolekcijai, formuojančiai pirmąją modelio erdvės iškarpančiąją kreivę.
	...		
<i>M</i>	<i>CRVPNT(N)</i>	Rodyklė	Rodyklė į paskutiniąją modelio erdvės kreivę (<i>D</i> sekcijoje), naudojamą šioje ribojančioje esybėje.

$1+M$	$SENSE(N)$	Sveikasis	Orientacijos vėliavėlė paskutiniajai modelio erdvės kreivei, naudojamai ribos formavimui.
$2+M$	$K(N)$	Sveikasis	Asocijuotųjų parametrų erdvės kreivių kiekis, panaudotas sudaryti kolekcijai, formuojančiai paskutiniąją modelio erdvės iškarpančiąją kreivę.
$3+M$	$PSCPT(N,1)$	Rodyklė	Rodyklė į esybę D sekcijoje, kuri yra pirmoji susietoji parametrų erdvės kreivė, naudojama kolekcijai, formuojančiai paskutiniąją modelio erdvės iškarpančiąją kreivę.
...			
$2+K(N)+M$	$PSCPT(N,K(N))$	Rodyklė	Rodyklė į esybę D sekcijoje, kuri yra paskutinioji susietoji parametrų erdvės kreivė, naudojama kolekcijai, formuojančiai paskutiniąją modelio erdvės iškarpančiąją kreivę.

8.1.7. Kreivė parametriniame paviršiuje (142)

Aprašymas: Ši esybė susieja duotą paviršių ir kreivę kaip esančią ant to paviršiaus. Tegul

$$S = S(u, v) = (x(u, v), y(u, v), z(u, v))$$

reguliarus parametrinis paviršius, kurio sritis

$$D = \{(u, v) \mid u_1 \leq u \leq u_2 \text{ ir } v_1 \leq v \leq v_2\}.$$

Tegul $B = B(t)$ yra kreivė:

$$B(t) = (u(t), v(t)), \text{ su visais } a \leq t \leq b,$$

su reikšmėmis srityje D .

Kreivė $C_C(t)$ ant paviršiaus $S(u, v)$ yra dviejų projekcijų (S ir B) kompozicija:

$$\begin{aligned} C_C(t) &\stackrel{\Delta}{=} S \circ B(t) \\ &\stackrel{\Delta}{=} S(B(t)) \\ &\stackrel{\Delta}{=} S(u(t), v(t)) \\ &\stackrel{\Delta}{=} (x(u(t), v(t)), y(u(t), v(t)), z(u(t), v(t))), a \leq t \leq b. \end{aligned}$$

Kreivė B yra dvimatėje erdvėje, kuri apibrėžiama kaip paviršiaus S sritis. Dėl šios priežasties B kreivės išraiška, kuri išvesta iš kreivės paminėtos specifikacijoje, turi būti dvimatė (nuoroda iš *BPTR* rodyklės).

B esybės D sekcijos 9 lauko paskirties vėliavėlė nustatyta į „05“ (žr.: Lentelė 8.1.5), pažymint, jog B yra parametrinėje paviršiaus erdvėje. Taigi B mastelis negali būti keičiamas, taip pat jei B taikoma transformacijų matrica – projekcija privalo būti parametru srities D viduje.

Kreivė ant parametrinio paviršiaus aprašoma:

1. C_C projekcija ant paviršiaus $S(u, v)$.
2. B ir S projekcijų kompozicija duoda kreivę C_C .

Parametru sekcijoje saugomos trys rodyklės:

1. Rodyklė į kreivę, iš kurios išvesta $B(t)$.
2. Rodyklė į paviršių $S(u, v)$.
3. Rodyklė į tokia projekciją $C(r)$, kad:
 - $C_C(t)$ ir $C(r)$ turi tą patį atvaizdą modelio erdvėje.
 - $C_C(t)$ ir $C(r)$ turi tuos pačius pabaigos ir pradžios taškus.
 - Parametrus t ir r sieja matematinis ryšys.
 - $C_C(t)$ ir $C(r)$ turi būti tokie, kad t būtų susietas su r monotonišku reikšmės augimu.

D sekcijos parametrai: Parametru reikšmių tipas ir diapazonas apibrėžtas (žr.: Lentelė 8.1.8).

Nr.	Parametras	Reikšmės
1	Esybės tipo numeris	142
7	Transformacijų matrica	0, \Rightarrow
9a	Būsena: matomumas	00 – 01
9b	Būsena: priklausomybė	00 – 03
9c	Būsena: paskirtis	00
9d	Būsena: hierarchija	Ignoruojiama
13	Spalvos numeris	#, \Rightarrow
14	Forma	0

Lentelė 8.1.8: Kreivės ant parametrinio paviršiaus D sekcijos parametrai

P sekcijos parametrai:

<u>Nr.</u>	<u>Vardas</u>	<u>Tipas</u>	<u>Aprašymas</u>
1	<i>CTRN</i>	Sveikasis	Kreivės ant paviršiaus sukūrimo būdas: „0“ – neapibrėžta, „1“ – duotos kreivės projekcija ant paviršiaus, „2“ – dviejų paviršių susikirtimas, „3“ – izo-parametrinė kreivė (u arba v).
2	<i>SPTR</i>	Rodyklė	Rodyklė į D sekcijos įrašą, aprašantį paviršių S .
3	<i>BPTR</i>	Rodyklė	Rodyklė į D sekcijos įrašą, aprašantį kreivę B .

parametrinėje S erdvėje.

- | | | | |
|---|-------------|-----------|--|
| 4 | <i>CPTR</i> | Rodyklė | Rodyklė į D sekcijos įrašą, aprašantį kreivę C . |
| 5 | <i>PREF</i> | Sveikasis | Siunčiančios sistemos išraiškos prioritetas:
„0“ – neapibrėžta, „1“ – $S \circ B$,
„2“ – C , „3“ – vienodas prioritetas tiek $S \circ B$, tiek C . |

8.1.8. Apribotasis paviršius (143)

Aprašymas: Apribotojo paviršiaus esybė naudojama iškarpytam paviršiui aprašyti. Paviršius ir iškarpančiosios kreivės išreiškiamos parametriškai ir tenkina savybes D1 – D12, išvardintas 8.1.6 skyrelyje.

Galimi dviejų tipų aprašai. Jei $TYPE = 0$, tai pateikiamos nuorodos į paviršių ir jo ribas modelio erdvėje. Jei $TYPE = 1$, tai pateikiamos nuorodos į paviršių, jo ribas modelio erdvėje ir susietąsias parametrų erdvės kreivių kolekcijas kiekvienai iš iškarpančiųjų modelio erdvės kreivių, naudojamų ribom sudaryti. Dėl polių ir siūlių, susietosios parametrų erdvės kreivių kolekcijos, naudojamos riboms, nebūtinai yra uždaros parametrų erdvėje.

Apribotasis paviršius aprašomas, naudojant keletą esybių. Tai yra Apribotojo paviršiaus esybė (143 tipo), ribojančioji esybė (141 tipo), parametriškai išreikštos neiškarpytų paviršių esybės ir parametriškai išreikštos kreivės.

D sekcijos parametrai: Parametrų reikšmių tipas ir diapazonas apibrėžtas (žr.: Lentelė 8.1.9).

Nr.	Parametras	Reikšmės
1	Esybės tipo numeris	143
7	Transformacijų matrica	0, \Rightarrow
9a	Būsena: matomumas	00 – 01
9b	Būsena: priklausomybė	00 – 03
9c	Būsena: paskirtis	00
9d	Būsena: hierarchija	Ignoruojuama
13	Spalvos numeris	#, \Rightarrow
14	Forma	0

Lentelė 8.1.9: Apribotojo paviršiaus D sekcijos parametrai

P sekcijos parametrai:

<u>Nr.</u>	<u>Vardas</u>	<u>Tipas</u>	<u>Aprašymas</u>
1	<i>TYPE</i>	Sveikasis	Apribotojo paviršiaus išraiškos tipas: „0“ – ribojančios esybės turės nuorodas tik į modelio erdvės iškarpančiąsias kreives. Susietoji paviršiaus išraiška (rodyklė iš <i>STPR</i>) gali būti

			parametrinė.
			„1“ – ribojančios esybės turės nuorodas į modelio erdvės kreives bei į susietąsias parametrų erdvės kolekcijas. Susietasis paviršius (rodyklė iš <i>STPR</i>) parametrinės išraiškos.
2	<i>SPTR</i>	Rodyklė	Rodyklė į esybę, aprašančią neiškarpytą paviršių, kuri reiki apriboti, <i>D</i> sekcijoje. Jei <i>TYPE = 1</i> , tada paviršiaus išraiška bus parametrinė.
3	<i>N</i>	Sveikasis	Ribojančiųjų esybių kiekis.
4	<i>BDPT(1)</i>	Rodyklė	Rodyklė į pirmosios ribojančios esybės (<i>141 tipo</i>) įrašą <i>D</i> sekcijoje.
	...		
3+N	<i>BDPT(N)</i>	Rodyklė	Rodyklė į paskutiniosios ribojančios esybės (<i>141 tipo</i>) įrašą <i>D</i> sekcijoje.

8.1.9. Iškarpytasis (parametrinis) paviršius (144)

Aprašymas: Paprastoji uždaroji kreivė Euklido plokštumoje dalina plokštumą į dvi dalis: į apribotą ir neapribotą. Apribotoji vadinama vidiniu regionu, o neapribotoji – išoriniu. Iškarpytojo paviršiaus sritis apibrėžiama kaip išorinės ribojančios kreivės (ribos) vidinis regionas ir vidinių ribojančių kreivių (ribų) išorinis regionas, įskaitant ir pačias kreives. Iškarpytajam paviršiui naudojamos tos pačios projekcijos kaip ir neiškarpytam $S(u, v)$, tačiau šie paviršiai priklauso skirtingom sritim. Kreivės, apibrėžiančios išorines ar vidines iškarpytojo paviršiaus ribas, yra kreivės ant paviršiaus S ir perduodamos šiai esybei kaip kreivė ant parametrinio paviršiaus (*tipas 142*).

Tarkim, kad $S(u, v)$ yra reguliarus parametrinis paviršius, kurio sritis yra stačiakampis D , susidedantis iš taškų (u, v) , tokiu kad, $a \leq u \leq b$, $c \leq v \leq d$, kur $a < b$, $c < d$ duotos konstantos. Sakysim, kad S parametrinės reikšmės 3D Euklido erdvėje traktuojamos taip:

$$S = S(u, v) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix},$$

kiekvienam (u, v) taškui srityje D .

Taip pat S projekcijai taikomos tokios reguliarumo sąlygos:

- Turi tolydžius normalės vektorius srityje D .
- Projekcija yra vienas prie vieno srityje D .

- Pirmosios dalinės išvestinės vektoriai bet kuriam D taške tiesiškai nepriklausomi. Naudojami du paprastųjų uždarytųjų kreivių tipų apibrėžimai, leidžiantys aprašyti iškarpytojo (parametrinio) paviršiaus sritį.

- *Išorinė riba*: yra vienintelė esanti D srityje, tai gali būti ir pačią D sritį ribojanti kreivė.
- *Vidinė riba*: gali būti kelios arba nebūti iš viso. Vidinėm ribom būdingi du bruožai:
 - Vidinės ribos su savo vidiniais regionais tarpusavyje nepersidengia ir nesiliečia.
 - Kiekviena iš ribų yra vidiniame išorinės ribos regione.

Jei išorinė riba sutampa su D srities ribojančia kreive ir nėra vidinių ribų, tada iškarpytasis paviršius yra neiškarpytas.

D sekcijos parametrai: Parametrų reikšmių tipas ir diapazonas apibrėžtas (žr.: *Lentelė 8.1.10*).

Nr.	Parametras	Reikšmės
1	Esybės tipo numeris	144
7	Transformacijų matrica	0, \Rightarrow
9a	Būsena: matomumas	00 – 01
9b	Būsena: priklausomybė	00 – 03
9c	Būsena: paskirtis	00
9d	Būsena: hierarchija	Ignoruojama
13	Spalvos numeris	#, \Rightarrow
14	Forma	0

Lentelė 8.1.10: Iškarpytojo paviršiaus D sekcijos parametrai

P sekcijos parametrai:

<u>Nr.</u>	<u>Vardas</u>	<u>Tipas</u>	<u>Aprašymas</u>
1	<i>PTS</i>	Rodyklė	Rodyklė į paviršių, kurį reikia iškarpyti.
2	<i>N1</i>	Sveikasis	„0“ – išorinė riba sutampa su D srities išorine riba, „1“ – kitu atveju.
3	<i>N2</i>	Sveikasis	Kiekis paprastųjų uždarytųjų kreivių, apibrėžiančių vidines iškarpytojo paviršiaus ribas. Jei tokių nėra, tada $N2 = 0$.
4	<i>PT0</i>	Rodyklė	Rodyklė į kreivę, apibrėžiančią išorinę iškarpytojo paviršiaus ribą (arba „0“).
5	<i>PTI(1)</i>	Rodyklė	Rodyklė į kreivę, apibrėžiančią pirmąją vidinę iškarpytojo paviršiaus ribą, jei $N2 > 0$.

...
 4+N2 *PTI(N2)* Rodyklė Rodyklė į kreivę, apibrėžiančią paskutinįją vidinę iškarpytojo paviršiaus ribą.

8.1.10. Spalvos esybė (314)

Aprašymas: Šioje esybėje aprašoma spalva *RGB* paletės pagrindu. Atspalvis gaunamas, sudėjus visų trijų kanalų (*R* – raudona, *G* – žalia, *B* – mėlyna) intensyvumus. Maksimalios visų kanalų intensyvumo reikšmės atitinka baltą spalvą, minimalios – juodą. IGES bylą generuojanti programinė įranga 13-tam D sekcijos laukui priskirs vieną iš aštuonių standartinių reikšmių, vaizdiniai artimiausių šioje esybėje aprašomai spalvai.

D sekcijos parametrai: Parametrų reikšmių tipas ir diapazonas apibrėžtas (žr.: *Lentelė 8.1.11*).

Nr.	Parametras	Reikšmės
1	Esybės tipo numeris	314
7	Transformacijų matrica	Ignoruojama
9a	Būsena: matomumas	Ignoruojama
9b	Būsena: priklausomybė	00
9c	Būsena: paskirtis	02
9d	Būsena: hierarchija	Ignoruojama
13	Spalvos numeris	1 – 8
14	Forma	0

Lentelė 8.1.11: Spalvos esybės D sekcijos parametrai

P sekcijos parametrai:

<u>Nr.</u>	<u>Vardas</u>	<u>Tipas</u>	<u>Aprašymas</u>
1	<i>CC1</i>	Rodyklė	<i>R</i> kanalo intensyvumo reikšmė diapazone [0; 100]
2	<i>CC2</i>	Sveikasis	<i>G</i> kanalo intensyvumo reikšmė diapazone [0; 100]
3	<i>CC3</i>	Sveikasis	<i>B</i> kanalo intensyvumo reikšmė diapazone [0; 100]
4	<i>CCNAME</i>	Rodyklė	Spalvos pavadinimas (nėra būtinas).