

Computergrafiek

Bert De Saffel

Master in de Industriële Wetenschappen: Informatica Academiejaar 2018–2019

Inhoudsopgave

I	Theorie	3
1	Rastering	4
1.1	Inleiding	4
1.2	Rastering van rechte lijnen	4
1.2.1	Midpoint Subdivision	4
1.2.2	Assymetrische DDA	5
1.2.3	Symmetrische DDA	5
1.2.4	Kwadrant DDA	6
1.2.5	Octant DDA	6
1.2.6	Multistep DDA	6
1.3	Rastering van cirkels	7
1.4	Opvullen van veelhoeken	7
1.5	Voorstellen van dikke lijnen	7
1.6	Antialiasing	8
2	Transformaties en projecties	9
2.1	Affiene 3D Transformaties	9
2.1.1	Rotaties	9
2.1.2	Schaaloperaties en spiegelingen	10
2.1.3	Translaties	10
II	Examenvragen	11
3	Modelvragen eerste theorievraag	12
3.1	Rastering	12
3.2	Het algoritme van Bresenham	12
3.3	Rastering van veelhoeken en antialiasing	12
3.4	Transformaties	12
3.5	Projecties en clipping	13
3.6	Het algoritme van Cyrus-Beck	13
3.7	Clipping	13

4	Modelvragen tweede theorievraag	14
4.1	NURBS constructie van cirkels (§3.4.8, slides en lesnota's)	14
4.2	NURBS constructie van cirkels en lijnsegmenten (§3.4.8 en slides)	14
4.3	Reflectiemodellen (§5.2, behalve §5.2.1.2 en §5.2.1.3)	15
4.4	1D Wavelet transformaties	15
4.5	2D Wavelet transformaties (§4.5.1)	15
4.6	Toepassingen van wavelet transformaties	15
5	Informatie derde vraag	16

Deel I

Theorie

Hoofdstuk 1

Rastering

1.1 Inleiding

De inhoud van het vak computergrafiek behandelt het realistisch voorstellen van 3D objecten op een 2D uitvoerbeeld zoals een scherm of een blad papier (via een printer). Er zijn vier fasen in dit proces:

- **CAGD en NURBS:** Het beschrijven van een willekeurige kromme met zo weinig mogelijk parameters. De studie naar dit soort wiskundige stelsels wordt Computer Aided Geometric Design genoemd.
- **Rendering:** Het realistisch voorstellen van deze beschrijven van objecten, onafhankelijk van de plaats en kijkrichting van een virtuele waarnemer.
- **Perspectieve projectie:** Het transformeren en projecteren van de 3D structuur naar 2D opstelling.
- **Rastering:** Het selecteren van de juiste pixels die het best overeenkomen met deze 2D opstelling.

1.2 Rastering van rechte lijnen

Rastering is het proces om de juiste pixels te selecteren die het best overeenkomen met een bepaalde voorstelling. Dit proces wordt uitgevoerd door de uitvoercontroller van een grafische kaart. Een aandachtspunt van zo een grafische kaart is dat het moeilijk met floating point getallen kan werken, dus zullen algoritmen die enkel met gehele getallen werken de voorkeur krijgen. Volgende secties bespreken enkele algoritmen om een rechte lijn van 1 pixel dik voor te stellen. Uiteraard is voor elk algoritme vereist dat een begin en eindpunt bekend is.

1.2.1 Midpoint Subdivision

Dit is een recursieve methode die het midden van het fictieve lijnstuk neemt en de pixel selecteert die er het dichtste bij ligt. Daarna wordt deze procedure opnieuw uitgevoerd voor de twee helften van dit lijnstuk.

Voordelen

- Eenvoudig te implementeren

Nadelen

- Naarmate de hoek van de rechte 45 graden wordt, zal de selectie onregelmatig en dunner worden aangezien er voor dezelfde x-waarde meerdere y-waarden, en voor dezelfde y-waarde meerdere x-waarden mogelijk zijn.
- Dit algoritme werkt volledig met reële getallen.

1.2.2 Assymetrische DDA

DDA staat voor Digital Differential Analyzer. Dit soort algoritmen poogt het aantal reële getallen tot een minimum te beperken. Dit algoritme loopt het lijnstuk af met een stapgrootte van 1 pixel. Na elke stap wordt de pixel die het dichtst bij het lijnstuk ligt, geselecteerd.

Voordelen

- Vereist minder stappen dan de symmetrische variant.
- Het produceert een regelmatig patroon.

Nadelen

- Lijnen worden te dun getekend indien de hoek van de rechte 45 graden nadert.
- Dit algoritme gebruikt nog altijd floating point bewerkingen.

1.2.3 Symmetrische DDA

Dit algoritme loopt ook het lijnstuk af zoals assymetrische DDA, maar de stapgrootte is veel kleiner. Het produceert exact hetzelfde resultaat als Midpoint Subdivision.

Voordelen

- Wanneer de hoek 45 graden nadert, zal het ten opzichte van de assymetrische variant, een vollere lijn, maar met een minder regelmatig patroon selecteren.

Nadelen

- Sommige pixels worden meerdere keren geselecteerd aangezien de stapgrootte kleiner is als een pixel.
- Dit algoritme gebruikt nog altijd floating point bewerkingen.

1.2.4 Kwadrant DDA

Dit algoritme verhoogt ook de incrementen met 1 pixel, maar enkel in de x **of** y richting, nooit in beide richtingen tegelijk. Indien de vorige pixel boven de ideale lijn zit, zal x met 1 verhoogd worden. Als de vorige pixel onder de ideale lijn zit, zal y met 1 verhoogd worden. Het algoritme maakt gebruik van een discriminant Δ , die best ingesteld wordt op $\{(x_2 - x_1) - (y_2 - y_1)\}/2$. Op basis van Δ wordt er beslist of er naar rechts of boven geselecteerd moet worden. Indien Δ positief is, bevindt de pixel benadering zich te veel in de y-richting en wordt de nieuwe pixel in de x-richting getekend, en kan Δ verminderd worden met $(y_2 - y_1)$. Analoog, als Δ negatief is wordt de lijn verder getekend in de y-richting, en Δ verhoogd met $(x_2 - x_1)$.

Voordelen

- Dit algoritme vermijdt floating point getallen door de afstand van een pixel tot het lijnstuk te vermenigvuldigen met een factor. **ToDo: welke factor**

Nadelen

- Naarmate de hoek 45 graden nadert wordt de lijn **te dik** getekend aangezien de helft van de beslissingen naar boven selecteert, en de andere helft naar rechts.
- Een foute initialisatie van Δ kan ervoor zorgen dat pixels onterecht opschuiven.

1.2.5 Octant DDA

Dit algoritme verfijnt kwadrant DDA door overlappingen te vermijden. Dit wordt gerealiseerd door ook diagonale selecties toe te staan. Dit algoritme produceert nagenoeg hetzelfde resultaat als assymetrische DDA.

Voordelen

- Maakt geen gebruik van reële getallen.

Nadelen

- Naarmate de hoek 45 graden nadert wordt de lijn **te dun** getekend.

1.2.6 Multistep DDA

Moet niet gekend zijn

Dit algoritme probeert de snelheid van vorige algoritmen te optimaliseren, en heeft geen bijdrage tot het bijsturen van de output, ten opzichte van kwadrant DDA. Bij elke iteratie worden nu meerdere pixels geselecteerd in plaats van slechts één pixel. Een voorbeeld van zo een algoritme is double-step DDA. Dit algoritme bepaalt in één stap twee opeenvolgende pixels waardoor er keuze bestaat tussen vier patronen. In werkelijkheid levert dit algoritme een verhoogde prestatie van ongeveer 10% en zorgt er dan ook voor dat single-step methoden zelden gebruikt worden.

1.3 Rastering van cirkels

Het algoritme dat gebruikt wordt om cirkels te rasteren wordt doorgaans het *algoritme van Bresenham* genoemd. Dit algoritme is in staat om een volledige cirkel met straal R te rasteren op basis van enkel de cirkelsector met beginpunt $(0, R)$ en eindpunt $(\frac{R}{\sqrt{2}}, \frac{R}{\sqrt{2}})$, wat overeenkomt met $1/8$ van de totale oppervlakte van de cirkel. De andere sectoren kunnen inmiddels via symmetrie bekomen worden.

In elke iteratie wordt de x -waarde van de pixel verhoogd met 1. De verandering van de y -waarde is verbonden aan een voorwaarde die afhangt van een discriminant Δ_i . Deze wordt gedefinieerd als het verschil van de kwadraten van de afstanden van $y_{i-1}(S_i)$ en $y_{i-1} - 1(T_i)$ tot de cirkel.

$$\begin{aligned}\Delta_i &= \left[(x_{i-1} + 1)^2 + y_{i-1}^2 - R^2 \right] - \left[R^2 - (x_{i-1} + 1)^2 - (y_{i-1} - 1)^2 \right] \\ &= 2(x_{i-1} + 1)^2 + y_{i-1}^2 + (y_{i-1} - 1)^2 - 2R^2\end{aligned}$$

Indien $\Delta_i > 0$, dan wordt T_i geselecteerd, anders S_i . De discriminant bij een volgende iteratie Δ_{i+1} kan uit Δ_i berekend worden:

$$\begin{aligned}\Delta_{i+1} &= 2(x_{i-1} + 2)^2 + y_i^2 + (y_{i-1} - 1)^2 - 2R^2 \\ &= \Delta_i + 4x_{i-1} + 6 + y_i^2 - y_{i-1}^2 + (y_i - 1)^2 - (y_{i-1} - 1)^2 \\ &= \Delta_i + 4x_{i-1} + 6 \text{ als } \Delta_i \leq 0 \\ &= \Delta_i + 4(x_{i-1} - y_{i-1}) + 10 \text{ als } \Delta_i > 0\end{aligned}$$

De initieële waarde van Δ_1 kan bekomen worden door $x_{i-1} = 0$ en $y_{i-1} = R$ te substitueren: $\Delta_1 = 3 - 2R$. Dit algoritme maakt enkel gehele getallen.

1.4 Opvullen van veelhoeken

Het opvullen van een veelhoek gebeurt via een scanlijn, die zowel horizontaal als verticaal kan zijn. Voor elke x of y -waarde wordt er een lijn getrokken door de veelhoek en gaat men op zoek naar de intersecties, wat altijd een even aantal zal zijn. Alle intersecties met een even index worden opgevuld tot de volgende intersectie met een oneven index. Zonder aanpassingen zullen echter ook pixels geselecteerd worden die buiten de veelhoek liggen. Het algoritme moet aangepast worden zodat enkel inwendige pixels als intersectiepunten kunnen gekozen worden, ook al ligt een uitwendig pixel dicht bij de randlijn.

1.5 Voorstellen van dikke lijnen

Het verdikken van lijnstukken kan de gebreken van sommige DDA algoritmes maskeren. Er bestaan drie methodes om lijnen te verdikken:

1. **Replicatie:** Tijdens de selectie van de pixel worden ook nog pixels boven en onder deze pixel geselecteerd, afhankelijk van de op voorhand gespecificeerde dikte.
2. **Centreren:** Tijdens de selectie van de pixel wordt er rond deze pixel een vast aantal pixels geselecteerd (bv in de vorm van een vierkant). Dit heeft als nadeel dat pixels opnieuw kunnen geselecteerd worden, ook al zijn ze eerder geselecteerd.

3. **Opvulling:** Deze methode past het DDA algoritme tweemaal toe, waarbij het resultaat van deze twee uitvoeringen verschoven zijn. De pixels tussen deze twee objecten kunnen dan geselecteerd worden. Deze methode vraagt nog meer rekenwerk.

1.6 Antialiasing

Aliasing is het gevolg van het feit dat pixels slechts een discreet aantal posities kan innemen, waardoor er kartelingen ontstaan. Antialiasing is het proces om deze kartelingen tegen te gaan. Het antialiasingproces kan volgens twee categorieën gebeuren:

1. *Rastering en antialiasing gebeurd in twee fasen.* De eerste fase maakt gebruik van een DDA algoritme zoals die reeds besproken zijn. De tweede fase zal pas antialiasing toepassen. Hiervoor zijn er twee manieren:
 - (a) **Supersampling.** Deze methode gebruikt de vaststelling dat aliasing vermindert indien de resolutie opgedreven zou kunnen worden. Hierdoor wordt de rastering dan ook uitgevoerd in het geheugen van de computer, wat fijner is dan wat de hardware ter beschikking heeft. Elke pixel van een scherm kan bijvoorbeeld overeenkomen met 4, 9, 16 of 25 pixels. De intensiteit van een pixel hangt af van het aantal geheugenpixels dat geselecteerd worden tijdens het rasteringsproces. Deze methode is efficiënt bij het weergeven van opgevolde veelhoeken, maar bij dunne lijnen kan er niet altijd een verbeterend effect zijn.
 - (b) **Postfiltering.** Hier wordt het rasteringsproces normaal uitgevoerd. De antialiasingfase maakt gebruik van een rechthoek rond de huidige geselecteerde pixel. De intensiteit van deze pixel hangt af van de geselecteerde pixels die in deze rechthoek liggen. Hoe groter de rechthoek (3x3, 5x5, ...) hoe groter het effect van antialiasing.
2. *Gelijktijdige rastering en antialiasing.* De rastering en antialiasing gebeurd in éénzelfde stap. Eén van de algoritmen is het *Pitteway-Watkinson algoritme*. **ToDo: k**

Hoofdstuk 2

Transformaties en projecties

De computergrafiek maakt gebruik van twee families transformaties: *affiene transformaties* en *perspectieve projecties*. Affiene transformaties zoals rotaties, spiegelingen en translaties zijn belangrijk omdat ze collineariteit en parallelisme bewaren. Een projectie is een bijzondere vorm van een transformatie die toelaat om driedimensionale beelden voor te stellen op tweedimensionale apparaten.

2.1 Affiene 3D Transformaties

2.1.1 Rotaties

Er bestaan drie elementaire rotaties. Een punt met coördinaten (x, y, z) kan gerooteerd worden over de drie assen. De afbeelding van deze rotatie krijgt de coördinaten (x', y', z') . Afhankelijk van de gebruikte rotatieas wordt een andere matrix gebruikt om de originele kolomvector te vermenigvuldigen. Deze wordt dan ook de rotatiematrix $R_{x,\theta}$ genoemd.

- **Rotatie rond de x-as.**

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix}$$

- **Rotatie rond de y-as.**

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \cdot \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$

- **Rotatie rond de z-as.**

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \cdot \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Elke willekeurige rotatie waarbij de rotatieas door de oorsprong loopt, kan beschreven worden door 5 opeenvolgende elementaire rotatiematrices. Veronderstel een rotatieas u over een hoek β . Dit kan

bekomen worden door eerst de u -as te laten samenvallen met de x -as, gebruik makend van twee rotaties $R_{y,\theta}$ en $R_{z,-\phi}$. Daarna wordt een rotatie uitgevoerd rond de x -as over een hoek β , $R_{x,\beta}$. De laatste twee rotaties dienen om de eerste twee rotaties ongedaan te maken.

$$R_{u,\beta} = R_{y,\theta} \cdot R_{z,\phi} \cdot R_{x,\beta} \cdot R_{z,-\phi} \cdot R_{y,\theta}$$

Om na te gaan of een willekeurige matrix ook een rotatiematrix is moet het matrixproduct van de matrix met zijn getransponeerde $\mathbf{1}$ opleveren.

2.1.2 Schaaloperaties en spiegelingen

Een schaaloperatie kan eenvoudig gedefinieerd worden als :

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \cdot \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix}$$

Via deze schaalmatrix kunnen ook spiegelingen gedaan worden. Indien één van de schaalfactoren s_x , s_y of s_z -1 is, en de andere twee op +1, is er een spiegeling in een coördinatenvlak. Een spiegeling rond een rechte kan bekomen worden door 2 schaalfactoren op -1 te zetten en de overblijvende op +1. Een spiegeling rond de oorsprong kan bekomen worden door alle schaalfactoren op -1 te zetten. Elke rotatie kan ook beschouwd worden als een samenstelling van twee spiegelingen.

2.1.3 Translaties

Deel II

Examenvragen

Hoofdstuk 3

Modelvragen eerste theorievraag

Deze vraag wordt gequoteerd op 1/4 van de totaalpunten.

3.1 Rastering

1. Bespreek de verschillende algoritmen voor de *rastering van rechte lijnen*, zonder in detail in te gaan op *multi-step* varianten. Vermeld telkens hun voor- en nadelen. (§1.2 & §1.2.1)
2. Hoe kunnen de methodes aangepast worden om **dikke** lijnen voor te stellen ? (§1.5)

3.2 Het algoritme van Bresenham

1. Bespreek het doel van dit algoritme en geef de volledige uitwerking van het selectieproces. (§1.3)
2. Hoe kan deze methode aangepast worden om **dikke** cirkels voor te stellen ? (§1.5)

3.3 Rastering van veelhoeken en antialiasing

1. Hoe moeten algoritmen voor het rasteren van rechte lijnen gewijzigd worden indien men ze wil toepassen op het *opvullen van veelhoeken* ? (§1.4)
2. Geef het doel van *antialiasing*, het algemeen principe ervan, en drie algoritmen voor de praktische uitwerking (met voorbeelden). (§1.6)

3.4 Transformaties

1. Welke families transformaties worden in de computer-grafiek gebruikt, en waarom ?
2. Geef en bespreek de matrixrepresentaties van de verschillende types transformaties en hun samenstellingen. (§2.1 behalve §2.1.4)

3.5 Projecties en clipping

1. Welke soort projectie wordt in de computergrafiek gebruikt, en waarom ?
2. Leid de algemene matrixvorm van deze projectie af. (§2.2)
3. Wat is de bedoeling van clipping ? Bespreek clippen in twee en in drie dimensies. (§2.3 zonder deelparagrafen)

3.6 Het algoritme van Cyrus-Beck

1. Geef het doel, de toepasbaarheid, en de beperkingen van het algoritme, en de volledige uitwerking van het principe. Pas het algoritme stap-voor-stap toe op volgende viewport (*figuur wordt gegeven*) en een lijnstuk met eindpunten (§2.3.2)
2. Hoe clipt men meer ingewikkelde krommen en figuren ?

3.7 Clipping

1. Het algoritme van *Cohen-Sutherland*: geef het doel, de toepasbaarheid, en de beperkingen van het algoritme, en de volledige uitwerking van het principe. Pas het algoritme toe op relevante voorbeelden. Geef eveneens een variant van de techniek. (§2.3.1)
2. Het algoritme van *Sutherland-Hodgman*: geef het doel, de toepasbaarheid, en de beperkingen van het algoritme, en de volledige uitwerking van het principe. Pas het algoritme stap-voor-stap toe op volgend voorbeeld: (figuur wordt gegeven) . (§2.3.3)

Hoofdstuk 4

Modelvragen tweede theorievraag

Deze vraag wordt gequoteerd op 2/4 van de totaalpunten.

4.1 NURBS constructie van cirkels (§3.4.8, slides en les-nota's)

1. aMet welke *open-uniforme NURBS* van orde drie (graad twee) kun je een *halve cirkel* (met centrum in de oorsprong en straal 1) tekenen , zonder (reële) knooppunten met meervoudige multipliciteit te moeten gebruiken ? Geef de preciese locatie van de *controlepunten* (op een figuur), hun gewichten, en de corresponderende *knopenvector*. Uit hoeveel segmenten bestaat deze NURBS ?
2. Toon aan dat deze constructie inderdaad exact een halve cirkel oplevert.
3. Construeer van deze NURBS de *uniforme* representatie. Vermeld de conversiestappen om tot dit resultaat te bekomen. Waarom is de constructie van de uniforme representatie belangrijk ?

4.2 NURBS constructie van cirkels en lijnsegmenten (§3.4.8 en slides)

1. Met welke *NURBS* kun je exact een recht *lijnsegment* door twee punten tekenen ? Geef de preciese locatie van de *controlepunten* (op een figuur), hun gewichten, en de corresponderende *knopenvector*.
2. Met welke *NURBS* bestaande uit één enkel segment kun je een *halve cirkel* (met centrum in de oorsprong en straal 1) tekenen ? Geef de preciese locatie van de *controlepunten* (op een figuur), hun gewichten, en de corresponderende knopenvector. Wat is de graad van deze NURBS ?
3. Toon aan dat deze constructie inderdaad exact een halve cirkel oplevert.
4. Met welke *NURBS* bestaande uit één enkel segment kun je exact een *volledige cirkel* (met centrum in de oorsprong en straal 1) tekenen ? Geef de preciese locatie van de *controlepunten*

(op een figuur), hun gewichten, en de corresponderende *knopenvector*. Wat is de graad van deze NURBS ?

4.3 Reflectiemodellen (§5.2, behalve §5.2.1.2 en §5.2.1.3)

1. Waarom zijn reflectiemodellen noodzakelijk ?
2. Omschrijf het lokale reflectiemodel (Phong-model). Geef ondermeer de berekenings-voorschriften, de betekenis van de parameters, en de nadelen.
3. Geef en omschrijf (in het bijzonder de nadelen) van de drie mogelijke benaderingen voor de berekening van de *lichtintensiteit van zichtbare punten*, indien men het object beschrijft aan de hand van een verzameling *vlakke veelhoeken*.

4.4 1D Wavelet transformaties

1. Bespreek met behulp van *Multi-Resolutie-Analyse* de algemene concepten van wavelet transformaties. (§3.5.2)
2. Vertaal deze algemene concepten in het bijzonder geval van de *Haar-wavelet* transformatie. (§3.5.1 & §3.5.2)
3. Bespreek de noodzaak van *spline-wavelets* (1D). Wat is het verband tussen de *Haar-wavelet* transformatie en de *spline-wavelet* transformatie ? Geef een overzicht van de relatieve voor- en nadelen.
4. Beschrijf van lage orde 1D *open-uniforme* spline-wavelet transformaties de vorm van achter-eenvolgens de *schaalfuncties*, de *wavelets* en de *synthese filters*.

4.5 2D Wavelet transformaties (§4.5.1)

1. Bespreek de alternatieve methodes om 2D *schaalfuncties en wavelets* te construeren.
2. Beschrijf, aan de hand van *contourplotjes*, en voor elk van deze alternatieve methodes, de resulterende 2D *Haar-schaalfuncties en Haar-wavelets* van het laagste en het op één na laagste niveau.

4.6 Toepassingen van wavelet transformaties

1. Geef de meest relevante toepassingen in de computergrafiek van 1D *Haar-wavelet* en 1D *spline-wavelet* transformaties. (§3.5.4)
2. Geef de meest relevante toepassingen in de computergrafiek van 2D *Haar-wavelet* en 1D *spline-wavelet* transformaties. (§4.5.2)

Hoofdstuk 5

Informatie derde vraag

Deze vraag is een **oefening**, gequoteerd op 1/2 van de totaalpunten, over één of enkele van volgende onderwerpen:

- (1-3) de Casteljau constructie (van een punt met specifieke parameterwaarde) van een Bézier kromme
- (1) verhoging van de graad van Bézier splines (in één enkele stap); voorafgaand moet het verband tussen de *oude* en de *nieuwe* controlepunten opgesteld worden (vermenigvuldiging met een specifieke matrix, cfr. theorieles)
- (2) verhoging van de graad van Bézier splines (*stapsgewijs*: één graad verhogen per stap)
- (3,4) segmentering (subdivisie) van Bézier krommen (eventueel meerdere segmenten in één enkele stap)
- (9-10,14-16) constructie van de *kromtecirkel* in een punt van een Bézier kromme
- (5-7,9-11) constructie van de *Bézier representatie* van een (polynomiale) NURBS
- (6-10) constructie van controlepunten na toevoeging van één of meerdere *reële* knopen in de knopenvector van een (polynomiale) NURBS (zonder over te gaan op de Bézier representatie)
- (10) constructie van controlepunten na toevoeging van één of meerdere *virtuele* knopen in de knopenvector van een (polynomiale) NURBS (zonder over te gaan op de Bézier representatie)
- (11) berekening en constructie van de controlepunten van de *open-uniforme* representatie van een (polynomiale) NURBS met een *uniforme* knopenvector
- (12,19) berekening en constructie van de controlepunten van de *uniforme* representatie van een polynomiale of rationale NURBS met een *open-uniforme* knopenvector
- (12) de Boor constructie (van een punt met specifieke parameterwaarde) van een (polynomiale) NURBS
- (13) constructie van de *hodograaf* van een Bézier kromme of spline
- (13-16) vaststellen van de continuïteit in de knooppunten van Bézier splines (*stelling van Stürk*)

- (17) constructie van de controlepunten van de *uniforme* Lagrange representatie van een Lagrange geïnterpoleerde kromme met *niet-uniforme* knopenvector; schematisch aantonen hoe de berekening van de Bézier representatie van deze kromme zou kunnen uitgevoerd worden (ondermeer opstellen van de *inverse* van de Bézier basismatrix).
- (18,19) constructie van een *benadering door lijnstukken* van een uniforme NURBS door toepassing van het algoritme van Lane & Riesenfeld
- (20) constructie van een *triangulair schema* met behulp van het veralgemeend algoritme van Neville (voor een specifieke configuratie van inputgegevens), en berekening hieruit van de *gewichtsfuncties* en de *matrixrepresentatie*