FACULTEIT INGENIEURSWETENSCHAPPEN
EN ARCHITECTUUR

Computergrafiek - Examen

Bert De Saffel

Master in de Industriële Wetenschappen: Informatica Academiejaar 2018–2019

Gecompileerd op 20 februari 2019

UNIVERSITEIT
GENT

Inhoudsopgave

Hoofdstuk 1

Modelvragen eerste theorievraag

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

1.1 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)

   - **Midpoint subdivision.** Dit algoritme maakt gebruik van een recursieve procedure waarbij het midden van een lijnstuk berekent wordt, de overeenkomstige pixel selecteert, en daarna deze procedure opnieuw aanroept om de twee helften van het segment. Dit algoritme werkt nadelig naarmate een hellingshoek van 45 graden bereikt wordt. Het effect is dan onregelmatig en dunner. Het grootste nadeel is echter dat het me reële getallen moet uitgevoerd worden.
   - **Asymmetrische DDA.** Algoritmes van de familie DDA pogen het aantal reële getallen te beperken, en voeren hun berekeningen ook incrementeel uit in plaats van recursief. Bij asymmetrische DDA wordt de x-waarde wordt telkens met 1 verhoogt terwijl de y-waarde verhoogt wordt met de richtingscoëfficiënt van de rechte. In elke stap wordt een pixel berekend en geselecteerd. Indien de helling van de rechte met de x-as groter zou zijn dan 45 graden, worden de rollen van x en y omgewisseld. Hier is echter ook het probleem dat indien de hellingshoek 45 graden nadert, dat de lijn te dun wordt getekend.
   - **Symmetrische DDA.** De symmetrische variant van DDA vermijdt het onderscheid in functie van de hellingshoek ten opzichte van de x-as, door de incrementen in beide richtingen repetitief te halveren tot wanneer deze kleiner zijn dan een pixel. Nadat de incrementen berekend zijn, wordt hetzelfde principe toegepast als asymmetrische DDA. De x en y-waarde worden nu telkens verhoogt met de eerder berekende incrementen, en selecteren de juiste pixel. Aangezien de incrementen kleiner zijn dan een pixel, zullen sommige pixels meerdere malen geselecteerd worden, wat een impact heeft op de snelheid van het algoritme. Het resultaat indien de hellingshoek 45 graden nadert, is voller dan by de asymmetrische variant, maar het patroon is minder regelmatig. Het produceert hetzelfde resultaat als midpoint subdivision.
   - **Kwadrant DDA.** Dit algoritme verhoogt de incrementen met exact 1 pixel, maar nooit in beide richtingen tegelijkertijd. De verhouding van het aantal incrementen in de y-richting tot het aantal incrementen in de x-richting wordt bepaald door de richtingscoëfficiënt van de rechte. Er is een extra variabele, Δ die best geinitialiseerd wordt op $\frac{(x_2-x_1)-(y_2-y_1)}{2}$. Indien Δ foutief geinitialiseerd wordt, produceert dit algoritme slechte resultaten bij zeer

vlakke of zeer steile lijnstukken. De variabele Δ bepaalt of de lijn verder getekend moet worden in de x-richting, indien $\Delta \geq 0$, of in de y-richting wanneer $\Delta < 0$.

Een enorm voordeel van dit algoritme is dat het geen gebruik maakt van reële getallen, ook niet bij de variabele Δ aangezien het dubbele bijgehouden wordt. Bij een hellingshoek die 45 graden nadert, zijn de lijnstukken veel te dik. Dit is logisch aangezien de helft van de beslissingen het lijnstuk in de x-richting zullen selecteren, en de andere helft in de y-richting, telkens afwisselend van elkaar.

- **Octant DDA.** Dit laatste algoritme verfijnt kwadrant DDA door overlappende segmenten te vermijden. Indien de hellingshoek van het lijnstuk kleiner zou zijn dan 45 graden, zal octant DDA een pixel ofwel in de horizontale richting of in de diagonale richting selecteren, afhankelijk naargelang Δ positief of negatief is. Is de hellingshoek groter, dan zal octant DDA in de verticale of diagonale richting selecteren.

  Dit algoritme produceert ongeveer dezelfde resultaten als asymmetrische DDA, zonder gebruik te maken van reële getallen. Naarmate de hellingshoek 45 graden nadert, wordt het lijnstuk te fijn weergegeven.

2. Hoe kunnen de methodes aangepast worden om **dikke** lijnen voor te stellen? (§1.5)

   - **Replicatie.** Dit is een eenvoudige manier, dat bij elke iteratie van een DDA algoritme, enkele pixels boven en onder, of links en rechts van de geselecteerde pixels meeselecteren, afhankelijk van de gewenste dikte en de hellingshoek. De overgang tussen twee lijnstukken die op deze manier dikker gemaakt zijn, is niet altijd even goed. Deze techniek wordt enkel toegepast indien de lijnen relatief dun zijn.
   - **Replicatie met vorm.** Bij elke iteratie van een DDA algoritme wordt rond de huidige geselecteerde pixel, een vlak geselecteerd. Vaak gebruikte vlakken zijn vierkanten voor lijnstukken en cirkelvormige vlakken voor cirkels.
   - **Opvulling.** Een derde techniek is om tweemaal het DDA algoritme toe te passen, op lijnstukken die uit elkaar verschoven zijn. Daarna worden deze twee lijnstukken opgevuld met de techniek om veelhoeken op te vullen.

1.2 Het algoritme van Bresenham

1. Bespreek het doel van dit algoritme en geef de volledige uitwerking van het selectieproces. (§1.3)

   - Het algoritme van Bresenham is geschikt om pixels te rasteren in de vorm van een cirkel terwijl het ook enkel gehele getalllen vereist. Het algoritme berekent enkel de pixels voor de cirkelsector met beginpunt $(0, R)$ en eindpunt $(R/\sqrt{2}, R/\sqrt{2})$. De andere cirkelsectoren kunnen met behulp van symmetrie bekomen. Bij elke iteratie wordt de x-waarde van de pixel verhoogd. Het aanpassen van de y-waarde hangt af van de discriminant Δ_i. Is deze waarde strikt positief dan zal de y-waarde met 1 pixel verlaagt worden. Is de discriminant negatief, dan blijf de y-waarde dezelfde. De discriminant Δ_i wordt gedefinieerd als het verschil van de kwadraten van de afstanden van de twee mogelijke punten tot de cirkel:

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

De discriminant Δ_{i+1} kan uit Δ_i afgeleidt worden.

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

De beginwaarde van de discriminant, Δ_i kan bekomen worden door de substitutie $x_{i-1} = 0$ en $y_{i-1} = R$ toe te passen:

$$\Delta_1 = 3 - 2R$$

2. Hoe kunnen de methodes aangepast worden om **dikke** lijnen voor te stellen? (§1.5)

   - **Replicatie.** Dit is een eenvoudige manier, dat bij elke iteratie van een DDA algoritme, enkele pixels boven en onder, of links en rechts van de geselecteerde pixels meeselecteren, afhankelijk van de gewenste dikte en de hellingshoek. De overgang tussen twee lijnstukken die op deze manier dikker gemaakt zijn, is niet altijd even goed. Deze techniek wordt enkel toegepast indien de lijnen relatief dun zijn.
   - **Replicatie met vorm.** Bij elke iteratie van een DDA algoritme wordt rond de huidige geselecteerde pixel, een vlak geselecteerd. Vaak gebruikte vlakken zijn vierkanten voor lijnstukken en cirkelvormige vlakken voor cirkels.
   - **Opvulling.** Een derde techniek is om tweemaal het DDA algoritme toe te passen, op lijnstukken die uit elkaar verschoven zijn. Daarna worden deze twee lijnstukken opgevuld met de techniek om veelhoeken op te vullen.

1.3 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)

   - De veelhoek wordt afgetast met een horizontale of verticale scanlijnen. Voor elke scanlijn worden de intersectiepunten met de lijn en het veelhoek bepaalt. Het aantal intersectiepunten is altijd een even aantal. De intersecties worden (bij een horizontale scanlijn) gesorteerd op de x-waarde. Alle pixels x tussen een intersectie met een even index en een intersectie met oneven index, $x_{2i} \leq x \leq x_{2i+1}$, worden getekend. Om te voorkomen dat pixels buiten het veelvlak geselecteerd worden, beperkt men de selectie tot enkel inwendige pixels als intersectiepunten, ook als ligt een uitwendig pixel dichter bij de randlijn.

2. Geef het doel van *antialiasing*, het algemeen principe ervan, en drie algoritmen voor de praktische uitwerking (met voorbeelden). (§1.6)

   - Antialiasing is het proces om de discrete eigenschap dat pixels vertonen, namelijk het gekarteld uitzicht, te minimaliseren.
     (a) **Supersampling.** Deze techniek maakt gebruik van een interne geheugenbuffer. Deze buffer biedt een fijner raster aan dan die dat hardwarematig beschikbaar zijn. Een pixel op een computerscherm kan overeenkomen met 16 pixels in het geheugen. Na het rasteringsproces, dat nu uitgevoerd wordt in het geheugen, wordt voor elke pixel nagegaan hoeveel geheugenpixels in de groep geselecteerd zijn. Als 7 pixels geselecteerd zijn met een geheugenpixelgrootte van 16 pixels, dan zal die pixel een intensiteit van 7/16 bedragen.

(b) **Postfiltering.** Deze methode zal eerst het rasteringsproces uitvoeren in het geheugen, dat nu dezelfde resolutie heeft als de hardware. Nadien wordt de intensiteit van een pixel herberekend op basis van het al dan niet geselecteerd zijn van de naburige pixels. De groep dat in beschouwing wordt genomen heeft vaak de vorm van een rechthoek (meer specifiek, een vierkant). Het gewicht van een naburige pixel wordt berekend op basis van de afstand van die pixel tot het middenpunt van de groep.

(c) **Prefiltering.** In tegenstelling tot de vorige twee algoritmen zal het rasterproces en antialiasingproces zich in dezelfde stap plaatsvinden. Dit algoritme heeft nood aan aangepaste DDA algoritmen. Een specifieke implementatie is het Pitteway-Watkinson algoritme. Dit algoritme maakt het mogelijk om een pixel te selecteren terwijl zijn intensiteit incrementeel berekend wordt. ToDo: ??

1.4 Transformaties

1. Welke families transformaties worden in de computer-grafiek gebruikt, en waarom?

   - **Affiene transformaties.** Hieronder vallen de rotaties, schaaloperaties, spiegelingen en translaties. Deze transformaties behouden colliniariteit en parallellisme. Evenwijdige lijnen zullen na een affiene transformatie nog steeds evenwijdig zijn.
   - **Perspectieve projecties.** Deze transformaties maken het mogelijk om een driedimensionaal object voor te stellen op een tweedimensionaal uitvoerapparaat.
   - De combinatie van deze twee transformaties biedt de mogelijkheidt om onder andere:
     - een object vanuit een andere kijkrichting te bekijken.
     - hij construeren van een object dat uit meerdere identieke componenten bestaat.

2. Geef en bespreek de matrixrepresentaties van de verschillende types transformaties en hun samenstellingen. (§2.1 behalve §2.1.4)

   - **Rotaties.**

1.5 Projecties en clipping

1. Welke soort projectie wordt in de computergrafiek gebruikt, en waarom?

   - De computergrafiek maakt gebruik van perspectieve projecties. Deze soort projecties simuleren de werking van het menselijk oog. Er wordt een oogpunt en een kijkrichting vastgelegd. De kijkrichting is een georiënteerde rechte door het oogpunt. Tussen het oogpunt en elk object wordt een rechte verbonden. Ergens op die rechte zal er een snijpunt zijn met het projectievlak. Dit snijpunt bepaalt het getransformeerde beeld van het punt.

2. Leid de algemene matrixvorm van deze projectie af. (§2.2)

   - Een perspectieve projectie kan afgeleid worden na het toepassen van een affiene transformatie. Een translatie T verplaatst het snijpunt van de kijkrichting met het projectievlak naar de oorsprong van het coördinatenstelsel. Deze transformatie wordt gevolgd door een rotatie R die het oogpunt op de negatieve z-as plaatst, in het punt met homogene coördinaten (0, 0, -d, 1). Hierdoor valt het projectievlak samen met het xy-vlak. Hierdoor is ook de z-waarde van het geprojecteerd punt nul. Op dit moment kunnen twee driehoeken beschouwd worden. De driehoek met de kijkrichting vanuit de y-as naar het zx-vlak (1) en de driehoek met de kijkrichting vanuit de x-as naar het zy vlak (2).

(1) Hieruit kan afgeleidt worden dat $\frac{x_p}{d} = \frac{d}{z+d}$.
(2) Hieruit kan afgeleidt worden dat $\frac{y_p}{d} = \frac{d}{z+d}$.
In homogene coördinaten kan dit echter vereenvoudigd worden.:

$$x' \equiv w' \cdot x_p = x \qquad y' \equiv w' \cdot y_p = y \qquad \text{met} \qquad w' = \frac{z}{d} + 1$$

In matrixvorm:

$$\begin{pmatrix} x' \\ y' \\ 0 \\ w' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Hieruit is het duidelijk dat een perspectieve projectie ook kan berekend worden via eenvoudige matrixvermenigvuldigingen:

$$\begin{pmatrix} x' \\ y' \\ 0 \\ w' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{pmatrix} \cdot \begin{pmatrix} i_x & j_x & k_x & -\theta_x \\ i_y & j_y & k_y & -\theta_y \\ i_z & j_z & k_z & -\theta_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

3. Wat is de bedoeling van clipping? Bespreek clippen in twee en in drie dimensies. (§2.3 zonder deelparagrafen)

- Een kijkrichting is begrensd door een bepaalde maximumwaarde. Vanuit het oogpunt kan er een pyramide beschouwd worden. Elke pixel dat zich niet in deze pyramide bevindt moet geclipt worden, aangezien deze toch niet gezien kan worden. De doorsnede van de pyramide met het projectievlak wordt het viewport genoemd. De viewport is evenwijdig met de x- en y-assen met het centrum in de oorsprong van het coördinatenstelsel waarin het geprojecteerd beeld afgebeeld. De selectie van alle pixels in perspectieve projectie die binnen de viewport liggen noemt men clipping. 3D Objecten worden eerste geclipt vooraleer ze perspectief geprojecteerd wordt. 2D objecten worden geprojecteerd en daarna pas geclipt. Het clippen van lijnstukken en opgevulde veelhoeken wordt door vrijwel alle algoritmen ondersteund. Bij het clippen van complexere veelhoeken wordt eerst de omhullende veelhoek berekent, en worden alle zijden geclipt met de viewport.

1.6 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)

- Het algoritme van Cyrus-Beck is geschikt om lijnen te clippen ten opzichte van een willekeurige convexe veelhoek. Het algoritme maakt gebruik van de geparameteriseerde voorstelling van een lijnstuk: $P(t) = P_1 + t(P_2 - P_1)$. Indien $0 \leq t \leq 1$, dan bevindt P(t) zich op het lijnstuk. Het Cyrus-Beck algoritme spoort deze t-waarden op van de intersecties met de verschillende zijden van de viewport en hieruit afleiden ofdat het lijnsegment hetzij partieel, hetzij totaal binnen de viewport ligt.
  De eerste stap van dit algoritme berekent alle $n_i[P(t) - f_i]$ functies en de waarden van t_i waarvoor deze functies nul worden. Indien al deze functies niet-negatieve waarden hebben voor zowel $t = 0$ als $t = 1$, dan ligt het lijnstuk volledig binnen de viewport. Is dit slechts het geval voor $t = 0$ of $t = 1$, dan ligt het lijnstuk zeker partieel in de viewport. Er moet bijgevolg extra kandidaatsnijpunten met de viewport gezocht worden. Enkel t_j waarden waarvor $0 \leq t_j \leq 1$ komen hiervoor in aanmerking. Voor elke t_j waarde wordt nagegaan of $n_i \cdot [P(t_j) - f_i] \geq 0$. De kleinste en grootste t_j leveren de gezochte snijpunten op.

- Toegepast op een lijnstuk met punten $P_1(-1,1)$ en $P_2(9,3)$ in een viewport met dimensie 8 bij 4 (fig 2.23 p 27 in de cursus). Het lijnstuk wordt geparameteriseerd door

$$P(t) = \begin{cases} x = -1 + 10t \\ y = 1 + 2t \end{cases}$$

Neem $f_{L,B} = (0,0)$ en $f_{T,R} = (8,4)$. Uitrekenen van de $n_i[P(t) - f_i]$ functies wordt dan: Elke nulwaarde kan een waarde voor t zijn. Er moet echter gelden dat $n_i[P(t_j) - f_i] \geq 0, \forall i$.

zijde	$[P_1P_2]$	nulwaarde
$n_L, f_L = (0,0)$	$(-1+10t-0, 1+2t-0)\cdot(1,0) = -1+10t$	$1/10$
$n_R, f_R = (8,4)$	$(-1+10t-8, 1+2t-4)\cdot(-1,0) = 9-10t$	$9/10$
$n_B, f_B = (0,0)$	$(-1+10t-0, 1+2t-0)\cdot(0,1) = 1+2t$	$-1/2$
$n_T, f_T = (8,4)$	$(-1+10t-8, 1+2t-4)\cdot(0,-1) = 3-2t$	$3/2$

Dit komt neer op het substitueren van de nulwaarden t_i in elke andere functie en nagaan of dat het resultaat positief is voor elk van deze functies. In dit geval is dit enkel voor $t = 1/10$ en $t = 9/10$. Dit zijn dan ook de punten die snijden met de viewport.

2. Hoe clipt men meer ingewikkelde krommen en figuren?

- _ToDo: oplossen

1.7 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)

- Het algoritme van Cohen-Sutherland is geschikt voor het clippen van lijnstukken ten opzichte van een rechthoekig viewport. Dit algoritme is enkel aan te raden om 2D clipping toe te passen. Het algoritme wenst zo snel mogelijk te detecteren of lijnstukken volledig binnen of buiten het viewport liggen. Het algortime zal aan elk eindpunt van een lijnstuk een bitcode $b_3b_2b_1b_0$ toekennen. Bit b_0 wordt op 1 gezet indien het punt zich links van de viewport bevindt, en een bit b_1 identificeert een punt aan de rechterzijde van de viewport. Op dezelfde manier hebben punten onder en boven de viewport respectievelijk bits $b_2 = 1$ en $b_3 = 1$. Een lijnstuk dat volledig binnen de viewport ligt heeft dus twee eindpunten met bitcode 0000. De twee eindpunten van een lijnstuk AND'en kan twee gevallen opleveren:
  (a) Indien het resultaat verschillend van 0000 is, dan ligt het lijnstuk met zekerheid buiten de viewport.
  (b) Is het resultaat echter 0000 (zonder dat de bitcodes van beide eindpunten 0000 is) dan ligt slechts een deel van het lijnstuk in de viewport.

  In dit tweede geval wordt het midden van dit lijnstuk genomen, en wordt het algoritme recursief toegepast op beide helften. Deze iteraties moeten doorgevoerd worden tot wanneer de precisie van een pixel bereikt is. Het voordeel van deze methodiek is dat het performant in de hardware kan uitgevoerd worden. Er is enkel een deling door twee vereist en kan parallel uitgevoerd worden.

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)

- _ToDo: oplossen

Hoofdstuk 2

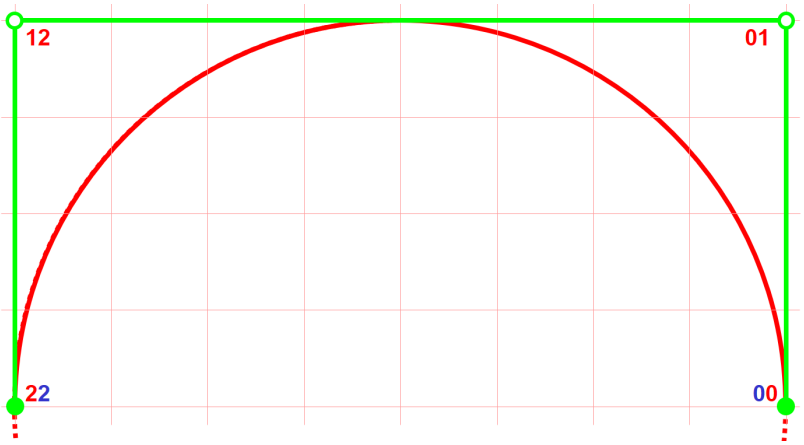
Modelvragen tweede theorievraag

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

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

1. Met 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 precieze locatie van de *controlepunten* (op een figuur), hun gewichten, en de corresponderende *knopenvector*. Uit hoeveel segmenten bestaat deze NURBS?

    - Maak gebruik van de gewichten $\{2, 1, 1, 2\}$ en knopenvector $\{0, 0; 0, 1, 2; 2; 2\}$. Deze NURBS bestaat uit twee segmenten. Figuur 2.1 illustreert de precieze locatie van de controlepunten en de twee segmenten (elk een kwart van de cirkelboog). De punten P_{00} en P_{22} hebben gewicht 2, terwijl de punten P_{12} en P_{01} een gewicht van 1 hebben.



Figuur 2.1: Open uniforme halve cirkel van orde drie.

2. Toon aan dat deze constructie inderdaad exact een halve cirkel oplevert.

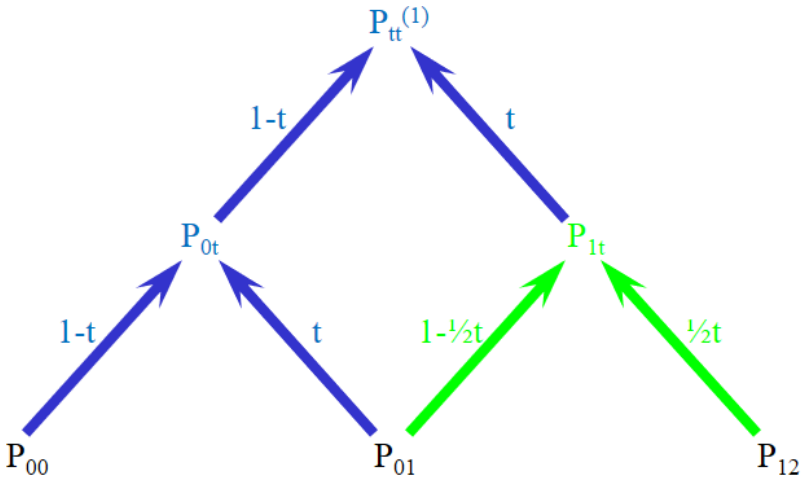
- Aangezien de cirkel een straal $R = 1$ heeft, ziet de vergelijking van de cirkel er als volgt uit:

$$\left(\frac{x}{w}\right)^2 + \left(\frac{y}{w}\right)^2 = 1$$

Na omvorming rest ons enkel te bewijzen dat

$$x^2 + y^2 - w^2 = 0$$

voor elk punt van de cirkel. Daarom stellen we het piramidaal schema op om een willekeurig punt $P_{tt}^{(1)}$ van het eerste segment te berekenen. Aangezien de twee segmenten symmetrisch zijn, moet dit niet opnieuw nagegaan worden voor een willekeurig punt van het tweede segment. Figuur 2.2 toont het schema dat gebruikt wordt. We weten dat



Figuur 2.2: Piramidaal schema voor het bewijs van de halve cirkel.

$$P_{00} = (1, 0)$$
$$P_{01} = (1, 1)$$
$$P_{12} = (-1, 1)$$

We kunnen deze punten echter voorstellen met homogene coördinaten, met w het gewicht van dat punt:

$$P_{00} = (2, 0, 2)$$
$$P_{01} = (1, 1, 1)$$
$$P_{12} = (-1, 1, 1)$$

Interpoleer de overeenkomstige punten P_{0t} en P_{1t}. Voor deze twee geïnterpoleerde punten bereken je onafhankelijk van elkaar de x, de y, en de w waarde:

$$P_{0t,x} = 2 * (1 - t) + 1 * t = 2 - t$$
$$P_{0t,y} = 0 * (1 - t) + 1 * t = t$$
$$P_{0t,w} = 2 * (1 - t) + 1 * t = 2 - t$$
$$P_{1t,x} = 1 * (1 - \frac{t}{2}) - 1 * \frac{t}{2} = 1 - t$$
$$P_{1t,y} = 1 * (1 - \frac{t}{2}) + 1 * \frac{t}{2} = 1$$
$$P_{1t,w} = 1 * (1 - \frac{t}{2}) + 1 * \frac{t}{2} = 1$$

Uiteindelijk kunnen deze twee punten opnieuw geïnterpoleerd worden om het punt $P_{tt}^{(1)}$ te bekomen.

$$\begin{aligned} P_{tt,x}^{(1)} &= (2-t)*(1-t)+(1-t)*t = 2-2t \\ P_{tt,y}^{(1)} &= t*(1-t)+1*t = -t^2+2t \\ P_{tt,w}^{(1)} &= (2-t)*(1-t)+1*t = t^2-2t+2 \end{aligned}$$

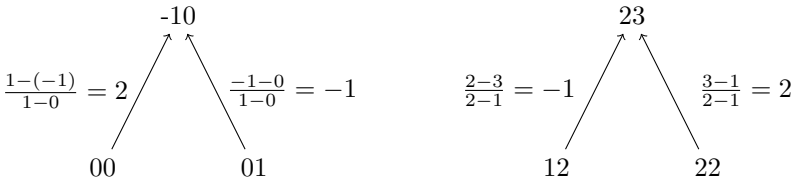
Nu dat we de x, y en w waarde van een punt kunnen bepalen voor eender welke waarde t, kan men nu de cirkelvergelijking invullen:

$$\begin{aligned} |P_{tt,x}^{(1)}|^2 + |P_{tt,y}^{(1)}|^2 - |P_{tt,w}^{(1)}|^2 &= (2-2t)^2 + (-t^2+2t)^2 - (t^2-2t+2)^2 \\ &= 4-8t+4t^2+t^4-4t^3+4t^2-t^4+4t^3-8t^2+8t-4 \\ &= t^4-t^4-4t^3+4t^3+4t^2+4t^2-8t^2+8t-8t+4-4 \\ &= 0 \end{aligned}$$

De opgegeven figuur is dus een exacte halve cirkel.

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?

- Elk punt $P_{00} = (1,0,1), P_{01} = (1,1,1), P_{12} = (-1,1,1)$ en $P_{22} = (-1,0,1)$ heeft een homogeen coördinaat met w het gewicht van die knoop. Deze punten zijn dus equivalent met $P_{00} = (2,0,2), P_{01} = (1,1,1), P_{12} = (-1,1,1)$ e $P_{22} = (-2,0,2)$. Verder kunnen we uit de open-uniforme knopenvector de uniforme knopenvector afleiden, aangezien die steeds oplopende getallen moet bevatten en de reëele knopen hetzelfde blijven: $\{-2,-1;0,1,2;3,4\}$. Aangezien het reëele deel hetzelfde blijft, verandert er aan de punten P_{01} en P_{12} niets. De punten P_{00} en P_{22} bestaan niet meer in de uniforme representatie, en worden vervangen door respectievelijk P_{-10} en P_{23}. Het verkregen piramidaal schema is te zien op figuur 2.3.



Figuur 2.3: Piramidaal schema om de uniforme representatie te construeren.

zodat

$$P_{-10} = 2P_{00} - P_{01} = 2*(2,0,2) - (1,1,1) = (3,-1,3)$$

en

$$P_{23} = -P_{12} + 2P_{22} = -(-1,1,1) + 2*(-2,0,2) = (1,-1,-1) + (-4,0,4) = (-3,-1,3)$$

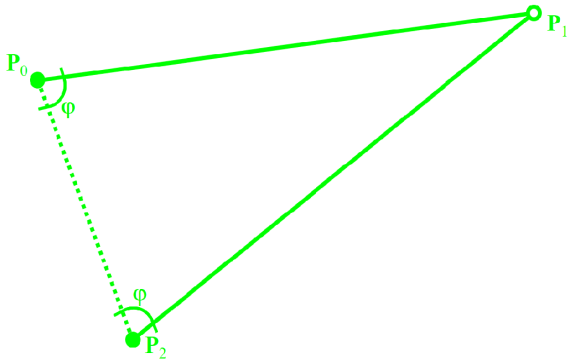
Uiteindelijk kan men de coördinaten bekomen door elk coördinaat terug om te zetten naar hun twee-dimensionale representatie, zodat $P_{-10} = (1,-\frac{1}{3}), P_{01} = (1,1), P_{12} = (-1,1)$ en $P_{23} = (-1,-\frac{1}{3})$ de nieuwe controlepunten worden.

- De constructie is belangrijk omdat, indien deze transformatie niet toegepast wordt, het algoritme van Lane en Riesenfeld niet kan toegepast worden. Dit algoritme is namelijk in staat om zeer efficiënt en eenvoudig knopen toe te voegen aan de B-spline, zonder dat de vorm van de B-spline wijzigt.

2.2 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 precieze locatie van de *controlepunten* (op een figuur), hun gewichten, en de corresponderende *knopenvector*.

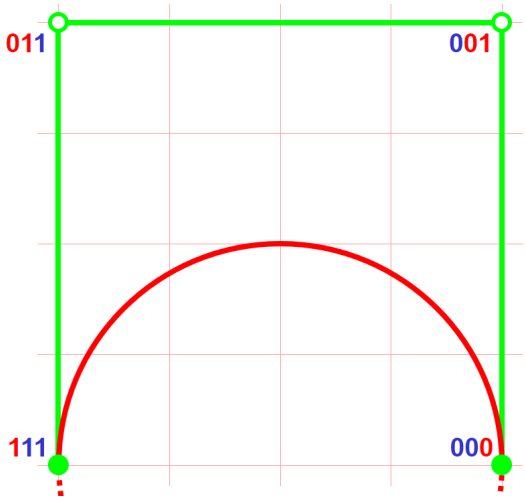
   - Figuur 2.4 toont hoe een lijnsegment door twee punten getekend kan worden. Dit is de controleveelhoek die gebruikt wordt om onder andere elliptische en parabolische segmenten te produceren. Indien de gewichtsfactor w gelijkgesteld wordt aan 0, krijgen we het lijnsegment $|P_0P_2|$. De knopenvector is $(0, 0; 0, 1; 1, 1)$.



Figuur 2.4: Recht lijnsegment door twee punten.

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

   - Figuur 2.5 toont de toont de precieze locatie van de controlepunten. De knopenvector is $(0, 0, 0; 0, 1; 1, 1, 1)$, de graad is 3 en de gewichten zijn $\{3, 1, 1, 3\}$.



Figuur 2.5: Een halve cirkel bestaande uit slechts één enkel segment met centrum in de oorsprong en straal 1.

3. Toon aan dat deze constructie inderdaad exact een halve cirkel oplevert.

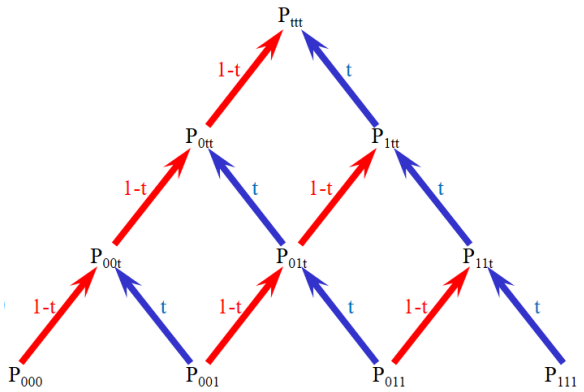
- Aangezien de cirkel een straal $R = 1$ heeft, ziet de vergelijking van de cirkel er als volgt uit:

$$\left(\frac{x}{w}\right)^2 + \left(\frac{y}{w}\right)^2 = 1$$

Na omvorming rest ons enkel te bewijzen dat

$$x^2 + y^2 - w^2 = 0$$

voor elk punt van de cirkel. Daarom stellen we het piramidaal schema op om een willekeurig punt P_{tt} van het segment te berekenen. Figuur 2.6 toont het schema dat gebruikt wordt. We weten dat



Figuur 2.6: Piramidaal schema voor het bewijs van de halve cirkel.

$$\begin{aligned} P_{000} &= (1, 0) \\ P_{001} &= (1, 2) \\ P_{011} &= (-1, 2) \\ P_{111} &= (-1, 0) \end{aligned}$$

We kunnen deze punten echter voorstellen met homogene coördinaten, met w het gewicht van dat punt:

$$\begin{aligned} P_{000} &= (3, 0, 3) \\ P_{001} &= (1, 2, 1) \\ P_{011} &= (-1, 2, 1) \\ P_{111} &= (-3, 0, 3) \end{aligned}$$

Uitvoering van het piramidaal schema:

$$\begin{aligned} P_{00t,x} &= 3(1-t) + 1t = 3 - 2t \\ P_{00t,y} &= 0(1-t) + 2t = 2t \\ P_{00t,w} &= 3(1-t) + 1t = 3 - 2t \\ P_{01t,x} &= 1(1-t) \ - \ 1t = 1 - 2t \\ P_{01t,y} &= 2(1-t) + 2t = 2 \\ P_{01t,w} &= 1(1-t) + 1t = 1 \\ P_{11t,x} &= -1(1-t) \ - \ 3t = -2t - 1 \\ P_{11t,y} &= 2(1-t) + 0t = 2 - 2t \\ P_{11t,w} &= 1(1-t) + 3t = 2t + 1 \end{aligned}$$

$$
\begin{aligned}
P_{0tt,x} &= (3-2t)(1-t) + (1-2t)t = 3-4t \\
P_{0tt,y} &= 2t(1-t) + 2t = 4t - 2t^2 \\
P_{0tt,w} &= (3-2t)(1-t) + 1t = 3 - 4t + 2t^2 \\
P_{1tt,x} &= (1-2t)(1-t) + \ (-2t-1)t = 1-4t \\
P_{1tt,y} &= 2(1-t) + (2-2t)t = 2 - 2t^2 \\
P_{1tt,w} &= 1(1-t) + (2t+1)t = 2t^2 + 1
\end{aligned}
$$

Uiteindelijk kunnen deze laatste twee punten opnieuw geïnterpoleerd worden om het punt P_{tt} te bekomen.

$$
\begin{aligned}
P_{ttt,x} &= (3-4t)(1-t) + t(1-4t) = 3-6t \\
P_{ttt,y} &= (4t-2t^2)(1-t) + t(2-2t^2) = 6t - 6t^2 \\
P_{ttt,w} &= (3-4t+2t^2)(1-t) + t(2t^2+1) = 3 - 6t + 6t^2
\end{aligned}
$$

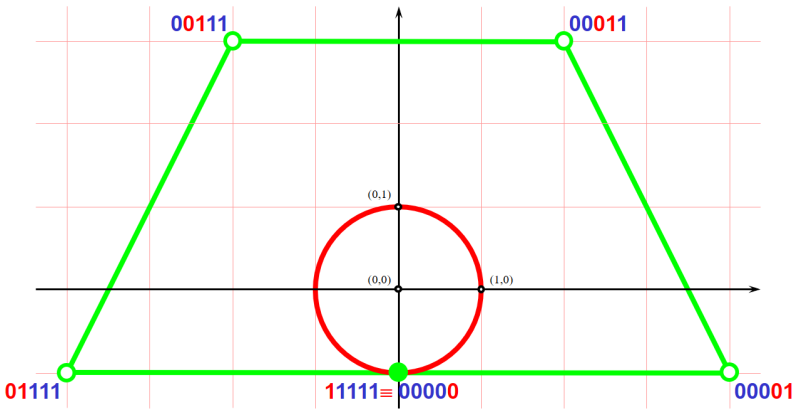
Nu dat we de x, y en w waarde van een punt kunnen bepalen voor eender welke waarde t, kan de cirkelvergelijking ingevuld worden:

$$
\begin{aligned}
|P_{tt,x}|^2 + |P_{tt,y}|^2 - |P_{tt,w}|^2 &= (3-6t)^2 + (6t-6t^2)^2 - (3-6t+6t^2)^2 \\
&= 9 - 36t + 36t^2 + 36t^2 - 72t^3 + 36t^4 - 36t^4 + 72t^3 - 72t^2 + 36t - 9 \\
&= 0
\end{aligned}
$$

De opgegeven figuur is dus een exacte halve cirkel.

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

   - Maak gebruik van de gewichten $\{5,1,1,1,1,5\}$ en knopenvector $\{0,0,0,0,0;0,1;1,1,1,1,1\}$. Figuur 2.7 illustreert de precieze locatie van de controlepunten. De graad van deze NURBS is 5. Punten P_{00000} en P_{11111} krijgen gewicht 5, de overige punten krijgen gewicht 1.



Figuur 2.7: Een volledige cirkel met slechts één segment van orde zes.

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

1. Waarom zijn reflectiemodellen noodzakelijk?

   - De precieze fracties geabsorbeerd, doorgelaten, diffuus teruggekaatst en spiegelend teruggekaatst licht hangen af van zeer veel factoren: ondermeer de invalshoek, kleur en intensiteit van het invallend lichtstraal, de gladheid van het oppervlak, en de samenstelling, temperatuur, kleur en transparantie van het object spelen een belangrijke rol. Deze fysische verschijnselen moeten benaderd worden door een mathematisch model, wat men het reflectiemodel noemt. Hoe meer gesofisticeerd dit reflectiemodel is, hoe realistischer de uiteindelijk weergegeven figuur er zal uitzien, maar ook hoe complexer en langduriger de vereiste berekeningen zijn.

2. Omschrijf het lokale reflectiemodel (Phong-model). Geef ondermeer de berekenings-voorschriften, de betekenis van de parameters, en de nadelen.

   - Een lokaal reflectiemodel houdt enkel rekening met de interactie tussen een aantal puntvormige lichtbronnen, het reflecterend oppervlak en het oogpunt. Met licht dat teruggekaatst of gebroken wordt door andere objecten wordt geen rekening gehouden. In het Phong-model berekent men voor elke RGB-component de intensiteit, uitgestuurd in de richting V van het oogpunt, als een som van drie bijdragen:

     $$I_{V,\text{rood}} = I_{D,\text{rood}} + I_{S,\text{rood}} + I_{G,\text{rood}}$$
     $$I_{V,\text{groen}} = I_{D,\text{groen}} + I_{S,\text{groen}} + I_{G,\text{groen}}$$
     $$I_{V,\text{blauw}} = I_{D,\text{blauw}} + I_{S,\text{blauw}} + I_{G,\text{blauw}}$$

     - I_D = Diffuse intensiteit.
       Deze bijdrage veronderstelt een ideaal mat oppervlak, en is bijgevolg enkel afhankelijk van de invallende lichtintensiteit I_L, de afstand R van het oogpunt tot het oppervlak, en de hoek θ tussen de invallende lichtstraal L en de normaal N op het oppervlak.

       $$I_D = k_D \sum_L I_L \frac{\cos\theta}{R^2} \qquad 0 \leq k_D \leq 1$$

     - I_S = Spiegelende intensiteit.
       Deze is het grootst in de richting R, die de teruggekaatste lichtstraal in de veronderstelling van een perfect reflecterend oppverlak weergeeft. Meestal neemt men een heuristische formule die dit gedrag vertoont, zoals,

       $$I_S = k_S \sum_L I_L \frac{\cos^n \Omega}{R^2} \qquad 0 \leq k_S \leq 1$$

       waarbij n het getal is dat de gladheid van het oppervlak karakteriseert en Ω de hoek tussen R en V.
     - I_G = Intensiteit ten gevolge van omgevingslicht.
       Dit wordt berekend als

       $$I_G = k_O \cdot I_O \qquad 0 \leq k_O \leq 1$$

       waarbij I_O een constante van de volledige figuur is, en k_O een materiaal afhankelijke coëfficiënt voorstelt.

     Vooral ten gevolge van de triviale benadering voor het omgevingslicht zijn lokale reflectiemodellen niet in staat om schaduwen weer te geven, en zien alle objecten er eerder als plastiek uit.

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*.

- **Uniform model:** Voor elk oppervlak wordt zijn normaal berekend die de lichtintensiteit voor het hele oppervlak bepaald. Het nadeel aan deze methode is dat elk punt van het oppervlak dezelfde intensiteit krijgt, wat zorgt voor een niet realistische voorstelling aangezien de verschillende segmenten, hoe klein ook, duidelijk zichtbaar zijn. Om dit te verhelpen wordt er in elk hoekpunt van het object een normaal N_A berekend, door de oppervlakte normalen van de aangrenzende segmenten uit te middelen.
- **Gouraud-model:** Op basis van de normalen N_A wordt eerst de lichtintensiteiten I_A berekent, van de hoekpunten van het object. Elk ander punt van een veelhoek van het object kan berekend worden door lineaire interpolatie ten opzichte van de lichtintensiteiten van de verschillende hoekpunten van de veelhoek. Een nadeel van deze methode is dat, als de normalen in de hoekpunten gelijk zouden zijn, maar het object toch kartelingen bevat, het object uniform gekleurd zou worden. Nog een tekortkoming is een gebrekkige weergave van spiegelende terugkaatsing. Het Gouraud-model is ook onderhevig aan het Mach-banden effect.
- **Phong-model:** Dit model interpoleert niet de hoekpunten zoals in het Gouraud-model, maar de normalen zelf. Pas hierna wordt de lichtintensiteit van elk punt berekend. Het model produceert normalen in een willekeurig punt van de benaderende veelhoek, die de normalen op het reële, gekromde vlak beter benaderen. Het Phong-model is wel wat rekenintensiever dan het Gouraud-model.

2.4 1D Wavelet transformaties

1. Bespreek met behulp van *Multi-Resolutie-Analyse* de algemene concepten van wavelet transformaties. (§3.5.2)

   - De multi-resolutie-analyse begint vanuit een rij lineaire functieruimten V^t. Deze functieruimten hebben als restrictie dat ze onderling genest moeten zijn:

     $$V^0 \subset V^1 \subset V^2 \subset V^3 \subset ...$$

     Een willekeurige functie $f^t(x)$ in V^t kan geschreven worden als een lineaire combinatie van de basisvectoren $\phi^t(x)$ van deze ruimte. Deze basisvectoren worden ook wel schaalfuncties genoemd.

     $$f^t(x) = \sum_{i=0}^{m_t-1} c_i^t \cdot \phi_i^t(x)$$

     Meestal groepeerd men deze schalingsfuncties in een rijmatrix:

     $$\Phi^t(x) = \begin{pmatrix}\phi_0^t(x) & \phi_1^t & ... & \phi_{m_t-1}^t(x)\end{pmatrix}$$

     Elke functie $f^{t-1}(x)$ kan ook geschreven worden als een lineaire combinatie van de basisvectoren van V^t. Aangezien Alle V^t onderling genest zijn, maakt $\phi^{t-1}(x)$ ook deel uit van V^t, met als gevolg dat elke basisvector $\phi_i^{t-1}(x)$ kan geschreven worden als een lineaire combinatie van de basisvectoren van V^t. Hiervoor wordt er een reductiematrix P^t gebruikt:

     $$\Phi^{t-1}(x) = \Phi^t \cdot P^t$$

     De reductiematrix P^t bevat als aantal rijen het aantal schalingsfuncties in de ruimte V^t, en als aantal kolommen het aantal schalingsfuncties in de ruimte V^{t-1}.
     Er zijn diverse keuzen voor de basisvectoren van V^t mogelijk: er kan gekozen worden om de schalingsfuncties van V^{t-1} aan te vullen met zogenaamde wavelets uit W^{t-1}. De wavelets uit W^t worden genoteerd als ψ_i^t, en worden ook in een rijmatrix gegroepeerd:

     $$\Psi^t(x) = \begin{pmatrix}\psi_0^t(x) & \psi_1^t(x) & ... & \psi_{n_t-1}^t(x)\end{pmatrix}$$

Deze wavelets moeten voldoen aan de volgende voorwaarde

$$< \phi_i^t(x) | \psi_j^t(x) > \equiv \int \big(\phi_i^t(x) \cdot \psi_j^t(x)\big) dx = 0, \qquad i \neq j$$

zodat de wavelets semi-orthogonaal zijn. De orthogonaliteit van wavelets vereist bovendien

$$< \psi_i^t(x) | \psi_j^t(x) > \equiv \int \big(\psi_i^t(x) \cdot \psi_j^t(x)\big) dx = 0, \qquad i \neq j$$

wat meestal niet voldaan is, maar wel voor Haar-wavelets. Aangezien nu de wavelets van W^{t-1} deel uitmaken van V^t, kan elke wavelet ϕ_i^{t-1} geschreven worden als een lineaire combinatie van de schaalfunctiers van V^t. Hiervoor wordt er een reductiematrix Q^t gebruikt:

$$\Psi^{t-1} = \Phi^t(x) \cdot Q^t$$

De reductiematrix Q^t heeft als aantal rijen het aantal schalingsfuncties in de ruimte V^t en als aantal kolommen het verschil van de aantallen schalingsfuncties in V^t en V^{t-1}.
Elke functie kan dus op diverse manieren gerepresenteerd worden. Als de combinatie $V^t = V^{t-1} + W^{t-1}$ gebruikt wordt, kan elke functie $f^{t+1}(x)$ als volgt geschreven worden:

$$f^{t+1}(x) = \sum_{i=0}^{m_t - 1} c_i^t \cdot \phi_i^t(x) + \sum_{i=0}^{n_t - 1} d_i^t \cdot \psi_i^t(x)$$

De coëfficiënten c_i^t en d_i^t worden in een rijmatrix voorgesteld:

$$C^t = \begin{pmatrix} c_0^t \\ c_1^t \\ \dots \\ c_{m_t-1}^t \end{pmatrix} \qquad D^t = \begin{pmatrix} d_0^t \\ d_1^t \\ \dots \\ d_{n_t-1}^t \end{pmatrix}$$

Op deze manier kan elke functie $f^{t+1}(x)$ nu geschreven worden als:

$$f^{t+1}(x) = \Phi^t(x) \cdot C^t + \Psi^t(x) \cdot D^t$$

De coëfficiëntenrijen C^{t-1} en D^{t-1} zijn equivalent met C^t. Om C^t te berekenen volstaat het om de alternatieve ontwikkeling in basisvectoren van een willekeurige functie te vergelijken:

$$\begin{aligned} \Phi^t(x) \cdot C^t = f^t(x) &= \Phi^{t-1}(x) \cdot C^{t-1} + \Psi^{t-1}(x) \cdot D^{t-1} \\ &= \Phi^t(x) \cdot P^t \cdot C^{t-1} + \Phi^t(x) \cdot Q^t \cdot D^{t-1} \\ &= \Phi^t(x) \cdot (P^t \cdot C^{t-1} + Q^t \cdot D^{t-1}) \end{aligned}$$

Zodat volgt dat

$$C^t = (P^t \cdot C^{t-1} + Q^t \cdot D^{t-1})$$

Om andersom, C^{t-1} en D^{t-1} te bepalen uit C^t introduceert men de analyse filters A^t en B^t:

$$C^{t-1} = A^t \cdot C^t \qquad D^{t-1} = B^t \cdot C^t$$

Hieruit kan afgeleidt worden dat de analyse en synthese (P^t en Q^t) filters elkaars geïnverteerden zijn.

$$\begin{aligned} \Phi^t(x) \cdot C^t = f^t(x) &= \Phi^{t-1}(x) \cdot C^{t-1} + \Psi^{t-1}(x) \cdot D^{t-1} \\ &= \Phi^{t-1}(x) \cdot A^t \cdot C^t + \Psi^{t-1}(x) \cdot B^t \cdot C^t \\ &= \Phi^t(x) \cdot P^t \cdot A^t \cdot C^t + \Phi^t(x) \cdot Q^t \cdot B^t \cdot C^t \\ &= \Phi^t(x) \cdot (P^t | Q^t) \cdot \begin{pmatrix} A^t \\ \overline{B^t} \end{pmatrix} \cdot C^t \end{aligned}$$

zodat

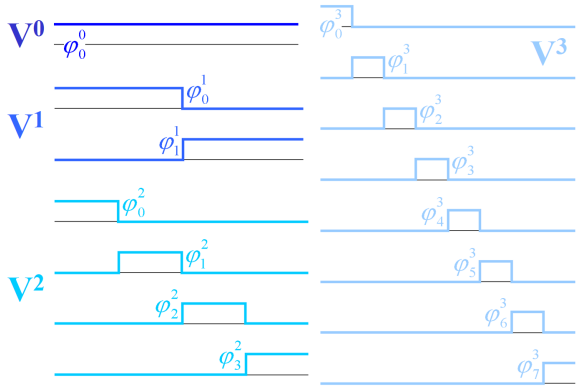
$$\left(\frac{A^t}{B^t}\right) = (P^t|Q^t)^{-1}$$

2. Vertaal deze algemene concepten in het bijzonder geval van de *Haar-wavelet* transformatie. (§3.5.1 & §3.5.2)

   - Elke niveau t benadering door een Haar-wavelet levert stuksgewijs constante functies op, constant in elk van de 2^t deelintervallen van het interval $[0, 1[$. Als Haar-schalingsfuncties neemt men meestal:

$$\phi_i^t(x) = \phi(2^t x - i) \quad , \quad \text{met} \quad \phi(x) = \begin{cases} 0 & , x < 0 \\ 1 & , 0 \leq x < 1 \\ 0 & , x \geq 1 \end{cases}$$

   Voorbeelden van de V^0, V^1, V^2 en V^3 schalingsfuncties zijn te zien op figuur 2.8. De



Figuur 2.8: De V^0, V^1, V^2 en V^3 Haar-schalingsfuncties.

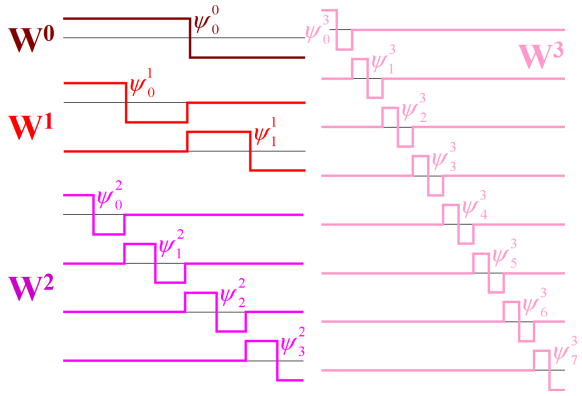
   reductiematrix P^t wordt dan gegeven door:

$$P^t = \begin{pmatrix} +1 & 0 & \dots & 0 & 0 \\ +1 & 0 & \dots & 0 & 0 \\ 0 & +1 & \dots & 0 & 0 \\ 0 & +1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & +1 & 0 \\ 0 & 0 & \dots & +1 & 0 \\ 0 & 0 & \dots & 0 & +1 \\ 0 & 0 & \dots & 0 & +1 \end{pmatrix}$$

   De Haar-wavelets zijn de wavelets die corresponderen met de Haar-schalingsfuncties en kunnen eveneens beschreven worden door:

$$\psi_i^t(x) = \psi(2^t x - i) \quad , \quad \text{met} \quad \psi(x) = \begin{cases} 0 & , x < 0 \\ +1 & , 0 \leq x < 1/2 \\ -1 & , 1/2 \leq x < 1 \\ 0 & , x \geq 1 \end{cases}$$

   Voorbeelden van de W^0, W^1, W^2 en W^3 Haar-wavelets zijn te zien op figuur 2.9. De wavelet $\psi(x)$ is de moederwavelet van de Haar-wavelet transformatie. Alle Haar-wavelets



Figuur 2.9: De W^0, W^1, W^2 en W^3 Haar-wavelets.

worden van de moederwavelet afgeleid door schaaloperaties en translaties. De reductiematrix Q^t van de Haar-wavelet transformatie wordt gegeven door de volgende eenvoudige matrix, bestaande uit 2^t rijen en $2^t - 1$ kolommen:

$$Q^t = \begin{pmatrix} +1 & 0 & ... & 0 & 0 \\ -1 & 0 & ... & 0 & 0 \\ 0 & +1 & ... & 0 & 0 \\ 0 & -1 & ... & 0 & 0 \\ ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... \\ 0 & 0 & ... & +1 & 0 \\ 0 & 0 & ... & -1 & 0 \\ 0 & 0 & ... & 0 & +1 \\ 0 & 0 & ... & 0 & -1 \end{pmatrix}$$

Voor de analyse filters van de Haar-wavelet transformatie bekomt men bijvoorbeeld volgende matrices met $2^t - 1$ rijen en 2^t kolommen:

$$A^t = \begin{pmatrix} +1 & +1 & 0 & 0 & ... & ... & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & +1 & ... & ... & 0 & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & 0 & ... & ... & +1 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & ... & ... & 0 & 0 & +1 & +1 \end{pmatrix}$$

$$B^t = \begin{pmatrix} +1 & -1 & 0 & 0 & ... & ... & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & -1 & ... & ... & 0 & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & 0 & ... & ... & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & ... & ... & 0 & 0 & +1 & -1 \end{pmatrix}$$

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.

   - De Haar-wavelets laten enkel stuksgewijs constante functies toe, en zijn door het ontbreken van continuïteit (zelfs geen C^0 continuïteit) te beperkt voor de meeste toepassingen in de computergrafiek, in het bijzonder voor de voorstelling van krommen en oppervlakken. Daarom zoekt men een betere techniek. Via de stelling van Curry en Schoenberg kan men aantonen dat elke B-spline die met een bepaalde knopenvector voorgesteld kan worden, ook kan voorgesteld worden door een willekeurig verfijnde knopenvector. Dit

heeft als gevolg dat alle mengfuncties behorend bij een bepaalde knopenvector, ook kunnen beschouwd worden als een lineaire combinatie van mengfuncties van een willekeurige verfijning van deze knopenvector. We kunnen dan de B-spline mengfuncties, behorend bij een rij onderling geneste knopenvectoren van een zelfde orde k, beschouwen als schaalfuncties van diverse niveau's van een spline-wavelet transformatie. In de praktijk wordt dit enkel gebruikt in combinatie met een open-uniforme knopenvector waarbij bovendien het aantal segmenten een macht van 2 is, 2^t, met t het niveau van de Multi-Resolutie-Analyse schalingsfunctie:

$$(0, ..., 0\ ;\ 0, 1, 2..., 2^t - 1, 2^t\ ;\ 2^t, ..., 2^t)$$

De virtuele knopen komen aan beide kanten k maal voor. Als spline-schalingsfuncties $\Phi^{t,k}$ heeft men de verzameling van $2^t + k - 1$ (het aantal controlepunten, waarvan 2 datapunten) mengfuncties $N_i(x) \equiv b_i^{t,k}(x)$, zoals die afgeleid kunnen worden met behulp van het algoritme van Cox en de Boor.

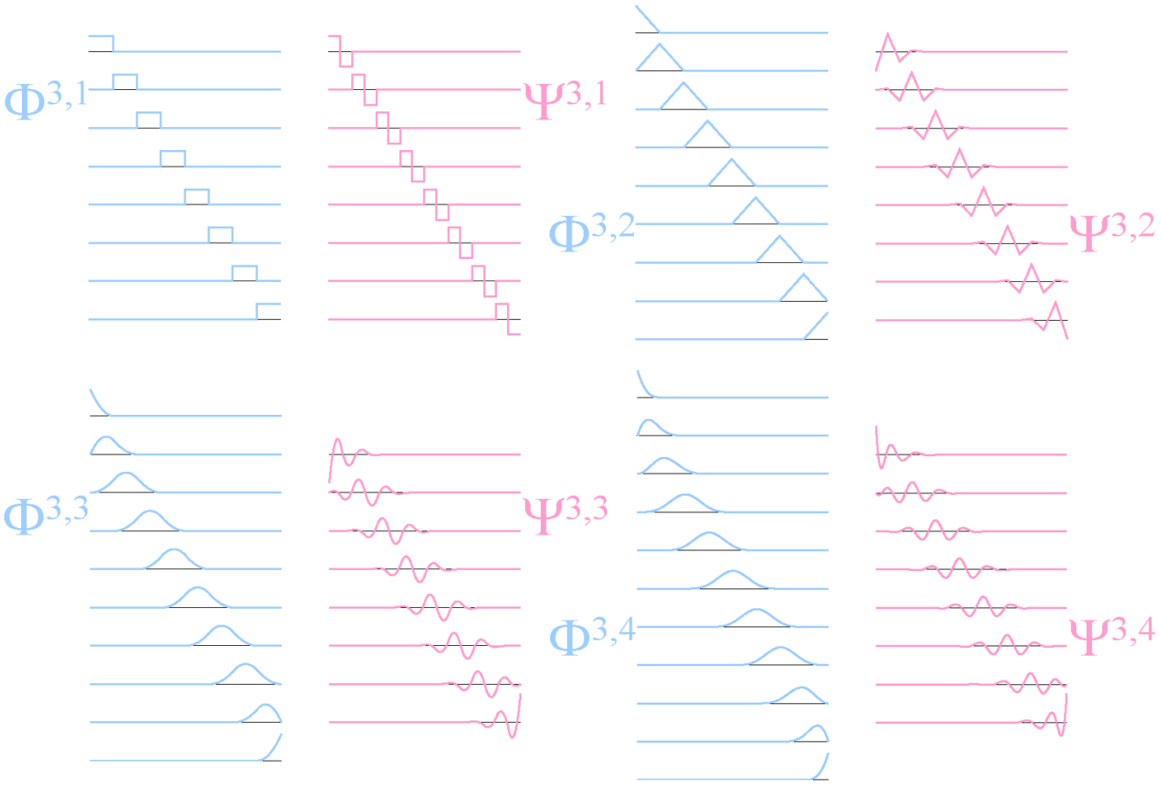
De Haar-wavelet transformatie is een specifieke vorm van de spline-wavelet transformatie met $k = 1$.

De relatieve voor- en nadelen zijn:

	Spline-wavelet transformatie		Haar-wavelet transformatie
+	Automatisch C^{k-2} continu	-	Zelfs geen C^0 continuïteit.
+	Veel toepassingen in de computergrafiek	-	Beperkte toepassing in de computergrafiek.
-	Voor alle $k > 1$ blijkt het enkel mogelijk te zijn om een verzameling semi-orthogonale spline-wavelets te construeren.	+	Altijd orthogonale wavelets.
-	Als gevolg van de semi-orthogonale spline-wavelets voor $k > 1$, kan men de analyse filters $A^{t,k}$ en $B^{t,k}$ slechts bekomen na expliciete numerieke inversie van de matrices samengesteld door de synthese filters $P^{t,k}$ en $Q^{t,k}$.	+	Bij Haar-wavelets kunnen de analyse filters bekomen worden door de synthese filters te transponeren.

4. Beschrijf van lage orde 1D *open-uniforme* spline-wavelet transformaties de vorm van achtereenvolgens de *schaalfuncties*, de *wavelets* en de *synthese filters*.

   - Figuur 2.10 illustreert de $\Phi^{3,k}$ spline-schalingsfuncties en $\Psi^{3,k}$ spline-wavelets voor $1 \leq k \leq 4$. De synthesefilters $P^{t,k}$ bestaan elk uit $2^t + k - 1$ rijen en $2^{t-1} + k - 1$ kolommen. De eerste $k-1$ en laatste $k-1$ kolommen hebben geen eenvoudige structuur. Het middendeel bevat voor elke kolom slechts $k+1$ niet-nul elementen, die op een constante factor na, de binomiaalcoëfficiënten blijken te zijn. Deze niet-nul elementen worden per opeenvolgende kolom telkens twee rijen naar onderen verschoven. De synthese filters $P^{3,k}$ voor $1 \leq k \leq 4$



Figuur 2.10: De $\Phi^{3,k}$ spline-schalingsfuncties en $\Psi^{3,k}$ spline-wavelets voor $1 \leq k \leq 4$.

worden hieronder in matrixvorm gegeven:

$$P^{3,1} = \begin{pmatrix} 1 & & & & \\ 1 & & & & \\ & 1 & & & \\ & 1 & & & \\ & & 1 & & \\ & & 1 & & \\ & & & 1 & \\ & & & 1 & \end{pmatrix} \qquad P^{3,2} = \frac{1}{2}\begin{pmatrix} 2 & & & & \\ 1 & 1 & & & \\ & 2 & & & \\ & 1 & 1 & & \\ & & 2 & & \\ & & 1 & 1 & \\ & & & 2 & \\ & & & 1 & 1 \\ & & & & 2 \end{pmatrix}$$

$$P^{3,3} = \frac{1}{4}\begin{pmatrix} 4 & & & & & \\ 2 & 2 & & & & \\ & 3 & 1 & & & \\ & 1 & 3 & & & \\ & & 3 & 1 & & \\ & & 1 & 3 & & \\ & & & 3 & 1 & \\ & & & 1 & 3 & \\ & & & & 2 & 2 \\ & & & & & 4 \end{pmatrix} \qquad P^{3,3} = \frac{1}{16}\begin{pmatrix} 16 & & & & & & \\ 8 & 8 & & & & & \\ & 12 & 4 & & & & \\ & 3 & 11 & 2 & & & \\ & & 8 & 8 & & & \\ & & 2 & 12 & 2 & & \\ & & & 8 & 8 & & \\ & & & 2 & 11 & 3 & \\ & & & & 4 & 12 & \\ & & & & & 8 & 8 \\ & & & & & & 16 \end{pmatrix}$$

De synthesefilters $Q^{t,k}$ bestaan uit $2^t + k - 1$ rijen en 2^t kolommen, en opnieuw worden in het middendeel van de matrix de niet-nul elementen per opeenvolgende kolom telkens twee rijen naar onder verschoven. De niet-nul elementen zijn groter in aantal $(3k - 1)$ en minder eenvoudig uit te drukken.

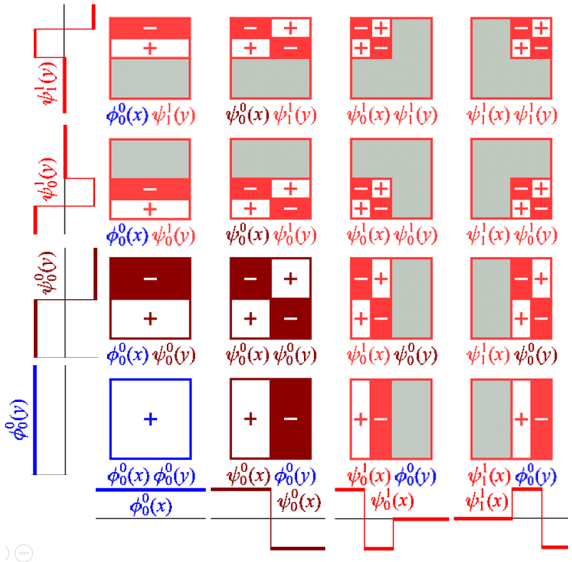
2.5 2D Wavelet transformaties (§4.5.1)

1. Bespreek de alternatieve methodes om 2D *schaalfuncties en wavelets* te construeren.

   - De 2D-schaalfuncties $\Phi^t(x, y)$ blijven beperkt tot producten van twee 1D-schaalfuncties die afhankelijk zijn van één enkele variabele. Deze vereenvoudiging maakt het mogelijk om corresponderende 2D-wavelets te construeren op basis van 1D-schaalfuncties en 1D-wavelets.
   - De 2D-wavelets kunnen op twee manieren geconstrueerd worden:
     (a) De standaard constructie berekent 2D-wavelets in een ruimte W^t, van willekeurige orde t, door alle mogelijke tensorproducten te nemen van alle 1D-wavelets in de ruimtes W^i van orde i, $0 \leq i \leq t$, aangevuld met de 1D-schaalfuncties van de functieruimte V^0.
     (b) De niet-standaard constructie neemt een aantal tensorproducten van 1D-wavelets in de ruimtes W^i van orde i, $0 \leq i \leq t$, aangevuld met 1D-schaalfuncties in de ruimtes V^i van orde i.

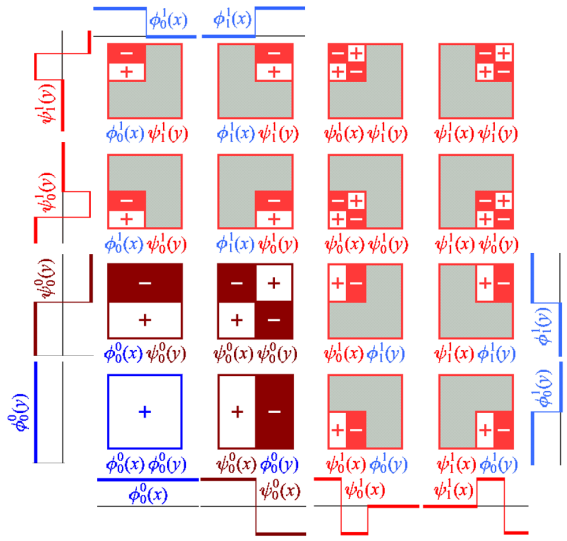
2. Beschrijf, aan de hand van *contourplotjes*, en voor elk van deze alternatieve methodes, de resulterende *2D Haar-schaalfuncties en Haar-wavelets* van het laagste en het op één na laagste niveau.

   - **Standaard constructie**: De standaard 2D-Haar wavelets in W^0 en W^1 kunnen bekomen worden door de tensorproducten te berekenen van de 1D-Haar-wavelets in W^0 en W^1, aangevuld met ϕ^0. De plustekens en mintekens duiden telkens de gebieden aan waar de wavelet +1 en -1 bedraagt. In de andere gebieden is de wavelet nul. Dit is te zien op figuur 2.11.



Figuur 2.11: De standaard constructie met behulp van contourplotjes.

   - **Niet-standaard constructie**: Men bekomt de niet-standaard 2D-Haar wavelets in W^1 door tensorproducten te berekenen van de 1D-Haar wavelets in W^0 en W^1, aangevuld met 1D-Haar schaalfuncties in V^0 en V^1. Dit is te zien op figuur 2.12



Figuur 2.12: De niet-standaard constructie met behulp van contourplotjes.

2.6 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)

   - **Beeldcompressie:** De Haar-wavelet is wel nuttig bij beeldcompressie. In deze toepassing beschouwt men een lineair beeld bestaande uit 2^t pixels als een functie in de ruimte V^t van Haar-schaalfuncties. De grijswaarden of kleurwaarden van de pixels bepalen de coëfficiënten C^t. Wavelet beeldcompressie voert een totale decompositie uit van een oorspronkelijk beeld, in termen van wavelets van verschillende niveau's $i (i < t)$, met coëfficiënten D^i, en de laagste niveau schaalfunctie. Termen met kleine wavelet coëfficiënten worden niet opgeslagen.
   - **Manipulatie van krommen en oppervlakken:** Een kromme wordt doorgaans beschreven als een gemiddelde van de coördinaten van controlepunten (of datapunten), gewogen met behulp van mengfuncties $B_i(x)$:

     $$P(x) = \sum_i B_i(x) \cdot P_i$$

     Beschouw nu de mengfuncties als de schaalfuncties $\Phi^j(x)$ van een ruimte V^j, $\Phi^j(x) \equiv \{B_i(x), \forall i\}$, en de coördinaten van de controlepunten als coëfficiënten C^j waarmee deze schaalfuncties moeten vermenigvuldigd worden, $C^j \equiv \{P_i, \forall i\}$, dan kan men hierop het principe van een filterbank toepassen. Door het recursief toepassen van de analyse filters A^t en B^t van een wavelet transformatie, bekomt men zowel coëfficiënten C^t van lagere niveau schaalfuncties $\Phi^t(x)$, als coëfficiënten D^t waarmee de corresponderende wavelets $\Psi^t(x)$ moeten vermenigvuldigd worden.
   - **Gladmaken van krommen:** Bij deze toepassing worden kwadratische en vooral kubieke spline-wavelets aangewezen. De spline-wavelet decompositie produceert zeer snel benaderende B-splines, bepaald door $2^t + k - 1$ controlepunten, met verschillende gehele niveau's t van gladheid.
   - **Reconstructie van 3D beelden:** Aan de hand van vlakke contourlijnen kunnen 3D beelden gereconstrueerd worden. Elke dwarse doorsnede bestaat meestal uit een kromme, bepaald door een groot aantal controlepunten. Het is de bedoeling elk controlepunt van

een dwarse doorsnede te verbinden met telkens twee opeenvolgende controlepunten van de volgende dwarse doorsneden. Op deze manier ontstaat tenslotte een driedimensionaal net van onderling met driehoeken verbonden controlepunten, dat het weer te geven oppervlak beschrijft. De complexiteit van het probleem wordt stapsgewijs vereenvoudigd door eerst opeenvolgende dwarse doorsneden volledig met behulp van een lineaire splinewavelet transformatie te reduceren. In de benadering van laagste niveau is het verbinden van de corresponderende controlepunten veel meer triviaal. Vervolgens wordt het oorspronkelijk beeld stapsgewijs gereconstrueerd, waarbij in elke stap triangulaties tussen de nog niet verbonden controlepunten en opeenvolgende dwarse doorsneden toegevoegd worden.

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

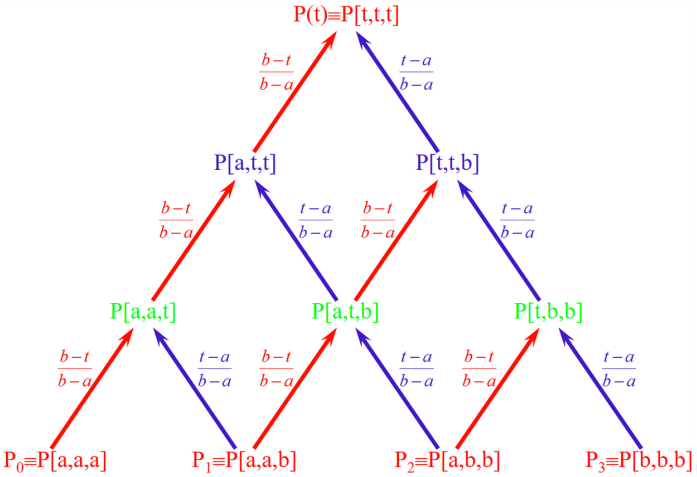
   - **Compressie en reconstructie:** Bij kleurbeelden wordt de compressie uitgevoerd op de diverse kleurcomponenten, onafhankelijk van elkaar. De kleinste afwijking tussen het oorspronkelijk beeld en het geconstrueerde beeld, kan bekomen worden door systematisch de wavelets met de kleinste coëfficiënten te verwaarlozen, ongeacht hun orde.
   - **Indexering:** Eerst wordt het te indexeren beeld met een Haar-wavelet transformatie gereduceerd. Vervolgens weerhoudt men enkel de coëfficiënten, in absolute waarde groter dan een bepaalde drempelwaarde. Tenslotte houdt men enkel rekening met het teken van de coëfficiënten. Een opzoeking kan als input een schets bevatten van het oorspronkelijke beeld. Deze input ondergaat dezelfde methode als om het beeld te indexeren. Vervolgens wordt de signatuur vergeleken met de signaturen in de databank. Beelden met de meeste overeenkomst komen in aanmerking als resultaat van deze opzoeking.
   - **Gladmaken van oppervlakken:** Net zoals 1D-wavelet transformaties worden 2D-wavelet transformaties ook aangewend voor het gladmaken van oppervlakken.
   - **Globale impact:** Controlepunten van een Bézier of NURBS oppervlak manipuleren heeft slechts een lokale impact. Als op dit oppervlak een 2D-wavelet decompositie uitgevoerd wordt, dan hebben die wijzigingen meer globaal karakter, naarmate het niveau lager is waarop men de wijzigingen uitvoert.
   - **Comprimeren:** Het biedt een eenvoudige manier om complexe oppervlakken in gecomprimeerde vorm op te slaan. het kan bijvoorbeeld gebruikt worden bij het doorsturen van beelden met een hoge resolutie over een netwerk met een relatief kleine bandbreedte. De zogenaamde *progressieve transmissie* voert een volledige reductie van het oorspronkelijk beeld uit. Vervolgens wordt eerst de laagste orde benadering doorgestuurd, en vervolgens de diverse wavelet coëfficiënten, in volgorde van stijgende resolutie. Van zodra de V^0 benadering is ontvangen, kan al een model afgebeeldt worden. Achtereenvolgens kan telkens een hogere benadering doorgestuurd worden.

Hoofdstuk 3

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
  - Algemeen kan een de Casteljau constructie zoals op figuur 3.1 berekent worden. Stel nu

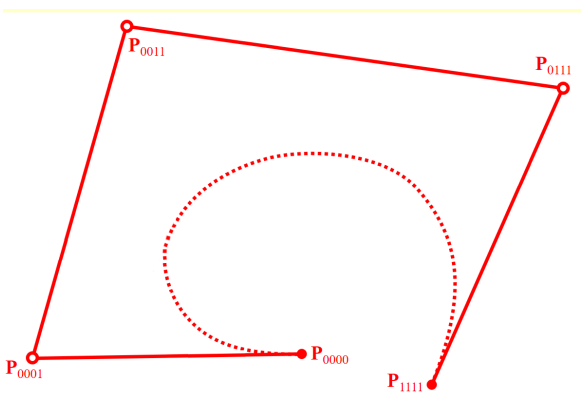


Figuur 3.1: Algemene methode de Casteljau constructie.

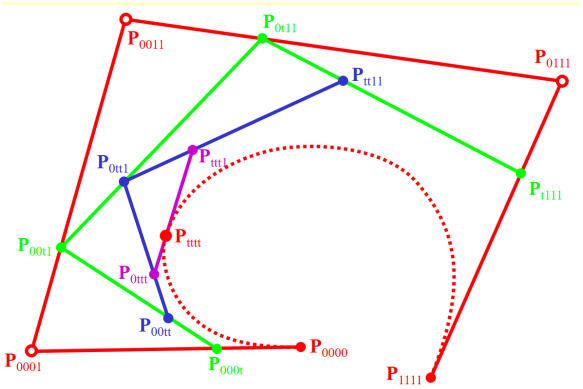
    dat hij de kromme van figuur 3.2 geeft. In dit geval komt 0 overeen met a en 1 overeen met b. Hij zal enkel nog een willekeurige t waarde geven. Vul het piramidaal schema in met deze t, en dan bekom je een gelijkaardige constructie zoals figuur 3.3.

- (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)
  - Eerst algemene formule (niet te kennen):

$$p[u_1, u_2, ...u_m] = \sum_{i_1=1}^{m} \sum_{\substack{i_2=1 \\ i_2 \neq i_1}}^{m} ... \sum_{\substack{i_n=1 \\ i_n \notin \{i_1,...i_n\}}}^{m} P[u_{i1}, u_{i2}, ..., u_{in}] \Big/ \binom{m}{n}$$



Figuur 3.2: Een Bézier kromme.



Figuur 3.3: de Casteljau constructie voor een willekeurige Bézier kromme.

Hierbij is n de oude graad en m de nieuwe graad waarbij geldt dat $m > n$. Een verhoging van de graad kan eenvoudig uitgevoerd worden door een matrix met $n+1$ kolommen en $m+1$ rijen op te bouwen. Als voorbeeld nemen we een Bézier kromme waarvan we de graad verhogen van 3 naar 6 (dus zeven punten). De matrix kan opgebouwd worden in drie stappen:

1. Een eerste hulpmatrix bevat de binomiale coëfficiënten waarbij een aantal restricties gelden: elke kolom mag slechts $m-n+1$ niet-nul elementen bevatten, waarbij elke volgende kolom steeds een rij lager begint. Ook moet rekening gehouden dat voor de laatste n rijen n_i, $(m-n+2 \leq i \leq m+1)$ de eerste $i-n-1$ kolommen vervangen moeten worden door een 0. Bij ons voorbeeld mag elke kolom slechts $6-3+1=4$ niet-nul elementen bevatten. Bovendien zal rij 5, 6 en 7 respectievelijk beginnen met 1, 2 en 3 nul-elementen.

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 3 & 3 & 1 \\ 0 & 4 & 6 & 4 \\ 0 & 0 & 10 & 10 \\ 0 & 0 & 0 & 20 \end{pmatrix}$$

2. Als tweede hulpmatrix nemen we de rotatie van 180 graden van de eerste matrix:

$$B = \begin{pmatrix} 20 & 0 & 0 & 0 \\ 10 & 10 & 0 & 0 \\ 4 & 6 & 4 & 0 \\ 1 & 3 & 3 & 1 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3. Nu kan de effectieve matrix opgesteld worden, door elk element A_{ij} te vermenigvuldigen met elk element B_{ij} (Geen matrixvermenigvuldiging, gewoon de individuele

elementen met elkaar vermenigvuldigen):

$$C = \begin{pmatrix} 20 & 0 & 0 & 0 \\ 10 & 10 & 0 & 0 \\ 4 & 12 & 4 & 0 \\ 1 & 9 & 9 & 1 \\ 0 & 4 & 12 & 4 \\ 0 & 0 & 10 & 10 \\ 0 & 0 & 0 & 20 \end{pmatrix}$$

Deze matrix kan nu eenvoudig vermenigvuldigt worden met de rijmatrix van de originele punten $P(t)$ om de nieuwe punten $Q(t)$ te bekomen. Elke kolom stelt één van de originele 4 punten voor. Een punt Q_i kan hieruit afgeleidt worden door op de i-de rij, de som van de producten van het originele punt met de factor voor die kolom te berekenen, en deze som te delen door $\binom{m}{n}$. In dit geval is $m = 6$ en $n = 3$, zodat $\binom{6}{3} = \frac{6!}{3!3!} = 20$. De zeven punten worden dus:

$$\begin{pmatrix} Q_{000000}(= Q_0) \\ Q_{000001}(= Q_1) \\ Q_{000011}(= Q_2) \\ Q_{000111}(= Q_3) \\ Q_{001111}(= Q_4) \\ Q_{011111}(= Q_5) \\ Q_{111111}(= Q_6) \end{pmatrix} = \frac{1}{20} \begin{pmatrix} 20 & 0 & 0 & 0 \\ 10 & 10 & 0 & 0 \\ 4 & 12 & 4 & 0 \\ 1 & 9 & 9 & 1 \\ 0 & 4 & 12 & 4 \\ 0 & 0 & 10 & 10 \\ 0 & 0 & 0 & 20 \end{pmatrix} \cdot \begin{pmatrix} P_{000}(= P_0) \\ P_{001}(= P_1) \\ P_{011}(= P_2) \\ P_{111}(= P_3) \end{pmatrix}$$

- $Q_0 = \frac{1}{20}(20P_0 + 0P_1 + 0P_2 + 0P_3) = P_0$
- $Q_1 = \frac{1}{20}(10P_0 + 10P_1 + 0P_2 + 0P_3) = \frac{1}{2}(P_0 + P_1)$
- $Q_2 = \frac{1}{20}(4P_0 + 12P_1 + 4P_2 + 0P_3) = \frac{1}{5}(P_0 + 3P_1 + P_2) = \frac{4}{5}(\frac{1}{4}P_0 + \frac{3}{4}P_1) + \frac{1}{5}P_2$
- $Q_3 = \frac{1}{20}(1P_0 + 9P_1 + 9P_2 + 1P_3) = \frac{1}{20}(P_0 + 9P_1 + 9P_2 + P_3)$
- $Q_4 = \frac{1}{20}(0P_0 + 4P_1 + 12P_2 + 4P_3) = \frac{1}{5}(P_1 + 3P_2 + P_3) = \frac{1}{5}P_1 + \frac{4}{5}(\frac{3}{4}P_2 + \frac{1}{4}P_3)$
- $Q_5 = \frac{1}{20}(0P_0 + 0P_1 + 10P_2 + 10P_3) = \frac{1}{2}(P_2 + P_3)$
- $Q_6 = \frac{1}{20}(0P_0 + 0P_1 + 0P_2 + 20P_3) = P_3$
- De punten Q_0 en Q_6 komen respectievelijk overeen met P_0 en P_3 en moeten niet gewijzigd worden.
- Voor de punten Q_1 en Q_5 passen we respectievelijk 1/2 af tussen P_0 en P_1 en tussen P_2 en P_3.
- Voor het punt Q_2 maken we eerst een hulppunt aan, dat op 1/4 van het punt P_1 ligt op het lijnstuk $[P_0P_1]$. Het punt Q_2 komt nu op 1/5 van R te liggen op het lijnstuk $[P_2R]$. Analoog voor het punt P_4.

- (2) verhoging van de graad van Bézier splines (*stapsgewijs*: één graad verhogen per stap)
  - Slechts één graad verhogen per stap is heel eenvoudig:

$$Q_k = \frac{k}{n+1}P_{k-1} + \frac{n+1-k}{n+1}P_k$$

Stel een bézierkromme van de 3de graad met punten $P_{000}, P_{001}, P_{011}$ en P_{111}.
De nieuwe punten van de bézierkromme van de 4de graad worden dan:

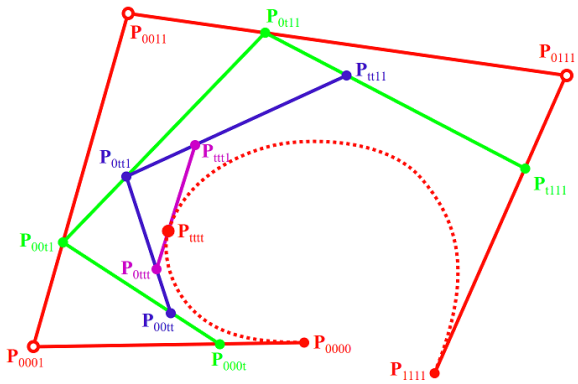
- $Q_{0000} = \frac{3+1-0}{3+1}P_{00} = P_{000}$
- $Q_{0001} = \frac{1}{4}P_{000} + \frac{4-1}{4}P_{001} = \frac{1}{4}P_{000} + \frac{3}{4}P_{001}$
- $Q_{0011} = \frac{1}{2}P_{001} + \frac{1}{2}P_{011}$
- $Q_{0111} = \frac{3}{4}P_{011} + \frac{1}{4}P_{111}$

◇ $Q_{1111} = P_{111}$

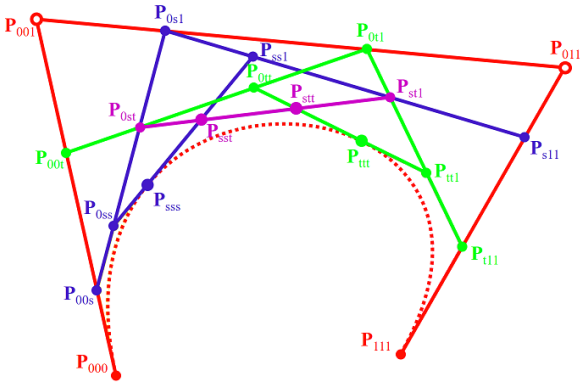
Het punt Q_{0001} ligt bijvoorbeeld op het punt waarvoor $t = 0.75$ op het lijnstuk $[P_{000}P_{001}]$.

- (3,4) segmentering (subdivisie) van Bézier krommen (eventueel meerdere segmenten in één enkele stap)

  ◦ Als je een de Casteljau constructie toepast op een Bézier kromme, introduceer je hulppunten, die perfect kunnen dienen als de nieuwe punten van twee of meerdere bézierkrommen. Bekijk figuur 3.4, waarbij de Bézierkromme gegeven wordt door de punten $(P_{0000}, P_{0001}, P_{0011}, P_{0111}, P_{1111})$. Het kan onmiddelijk gesegmenteerd worded in twee bézier krommen, respectievelijk bepaald door $(P_{0000}, P_{000t}, P_{00tt}, P_{0ttt}, P_{tttt})$ en $(P_{tttt}, P_{ttt1}, P_{tt11}, P_{t111}, P_{1111})$. Het is ook mogelijk om kubieke Bézierkrommen in meerdere segmenten op te splitsen, zoals te zien op figuur 3.5, die respectievelijk segmenten bepalen door $(P_{000}, P_{00s}, P_{0ss}, P_{sss})$, $(P_{sss}, P_{sst}, P_{stt}, P_{ttt})$ en $(P_{ttt}, P_{tt1}, P_{t11}, P_{111})$.



Figuur 3.4: Een bézierkromme van graad vier.



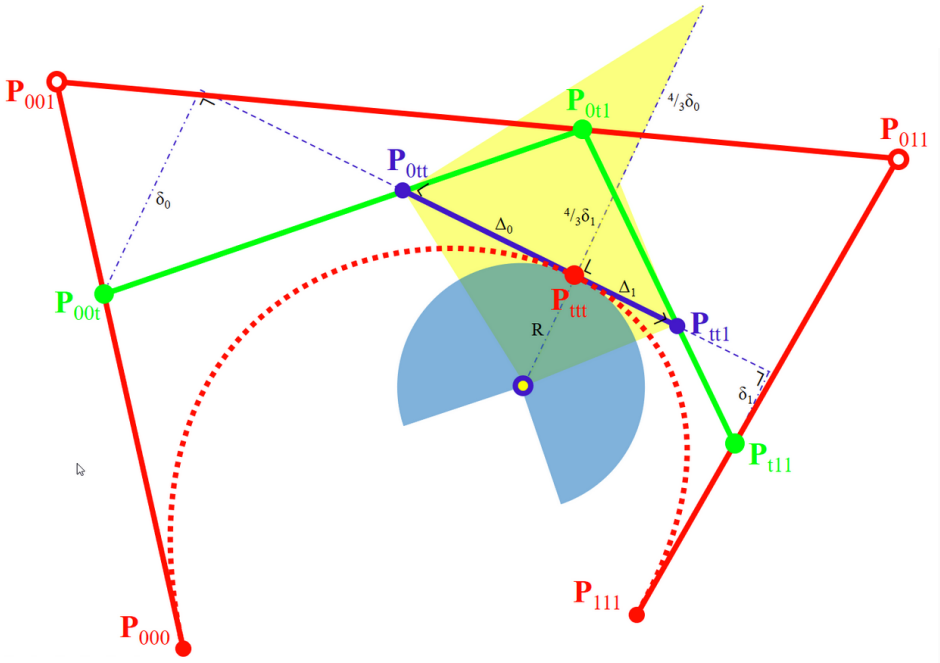
Figuur 3.5: Een kubieke bézierkromme van graad drie.

- (9-10,14-16) constructie van de *kromtecirkel* in een punt van een Bézier kromme

  ◦ Algemene methode om een kromtecirkel te berekenen in een punt van een Bézier kromme wordt weergegeven op figuur 3.6. De kromming van een Bézier kromme wordt bepaald door de hulppunten van de laatste 2 iteraties van het de Casteljau proces. Eerst moet dus een de Casteljau constructie uitgevoerd worden op de Bézier kromme voor het opgegeven punt t. De straal R van de kromtecirkel wordt gegeven door

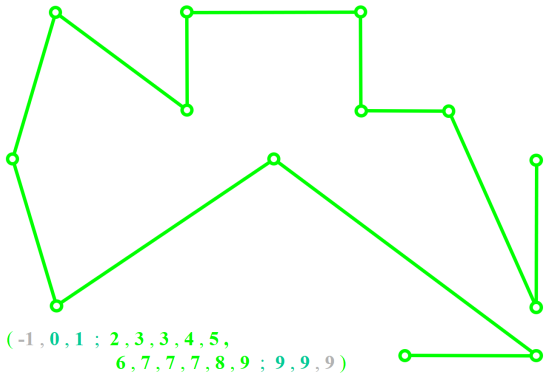
$$R = \frac{3\Delta^2}{4\delta}$$

Het volgende proces kan op twee manieren uitgevoerd worden, hier doe ik het voor de controlepunten (P_{00t}, P_{0tt}), maar zou evengoed gelden voor de controlepunten (P_{tt1}, P_{t11}). De afstand van P_{ttt} tot P_{0tt} noemen we Δ_0 (is vrij onbelangrijk, maar toch vermeld ik het omdat het op de figuur staat). Het lijnstuk dat samenvallend is met $|P_{ttt}P_{0tt}|$ moet doorgetrokken worden, zodanig dat er een rechte hoek gemaakt kan worden tussen het eindpunt van dit lijnstuk en het punt P_{00t}. De afstand van P_{00t} en dit eindpunt noemen we δ_0. Nu kunnen we een lijnstuk, loodrecht op het lijnstuk $|P_{0tt}P_{tt1}|$ en door het punt P_{ttt} met lengte $\frac{4}{3}\delta_0$ tekenen. Op dit moment kan er vanuit P_{0tt} een lijnstuk getrokken worden tot aan het einde van het vorige lijnstuk, gevolgd door een ander lijnstuk, loodrecht met het vorige lijnstuk, zodanig dat het kruist met het verlengde van het lijnstuk door P_{ttt}. Deze kruising is het middelpunt van de cirkel. De afstand van dit middelpunt tot P_{ttt} is de straal R van de cirkel.



Figuur 3.6: De kromtecirkel door het punt P_{ttt}.

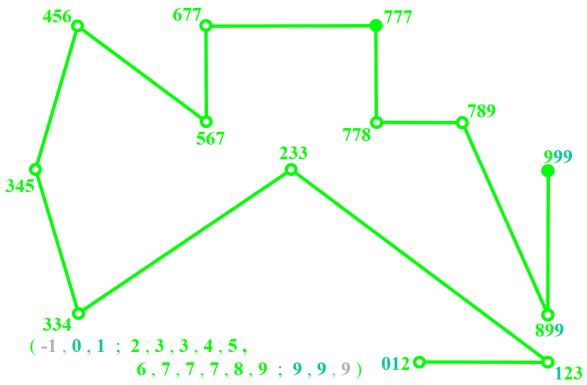
- (5-7,9-11) constructie van de *Bézier representatie* van een (polynomiale) NURBS
  - De knooppunten van de B-spline worden datapunten van de Bézier-representatie. Deze vraag wordt opgelost met het voorbeeld op figuur 3.7.



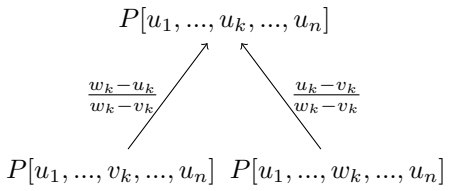
Figuur 3.7: Fase 1 bézier representatie van een polynomiale NURBS.

De knopenvector wordt gegeven: $(-1, 0, 1; 2, 3, 3, 4, 5, 6, 7, 7, 7, 8, 9; 9, 9, 9)$. Wat meteen opvalt is dat er 7 (waarvan sommige hogere multipliciteit) reële knooppunten zijn, wat betekent dat de resulterende bézier spline 6 segmenten zal bevatten. Schrijf eerst de blossomnotatie van elk knooppunt, zoals weergegeven op figuur 3.8.

Via de multilineariteit eigenschap kan men nieuwe punten introduceren. (herhaling op figuur 3.9)



Figuur 3.8: Fase 2 bézier representatie van een polynomiale NURBS.



Figuur 3.9: Multilineariteit eigenschap.

Bijvoorbeeld: tussen punt 012 en 123 moeten nog twee hulppunten komen: 112 en 122. Stel in het diagram $u_k = 1, v_k = 0$ en $w_k = 2$. dan krijgen we

$$P[1,1,2] = \frac{2}{3}P[0,1,2] + \frac{1}{3}P[1,2,3]$$

of anders gezegd: het punt 112 ligt op $t = 1/3$ voor het lijnstuk $[P[0,1,2]P[1,2,3]]$. Analoog voor $u_k = 2, v_k = 0$ en $w_k = 2$, dan krijgen we

$$P[1,2,2] = \frac{1}{3}P[0,1,2] + \frac{2}{3}P[1,2,3]$$

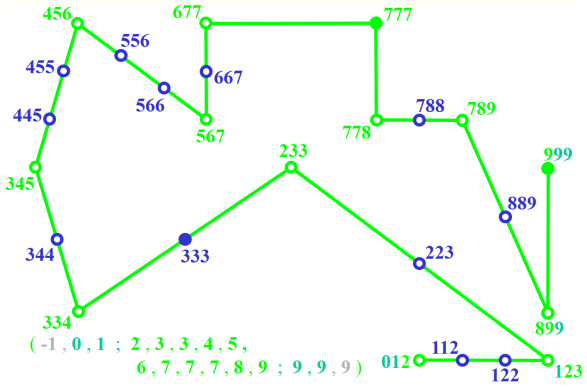
of anders gezegd: het punt 122 ligt op $t = 2/3$ voor het lijnstuk $[P[0,1,2]P[1,2,3]]$. Als je dit zou doen voor punten 123 en 233, dan zou je

$$P[2,2,3] = \frac{1}{2}P[1,2,3] + \frac{1}{2}P[2,3,3]$$

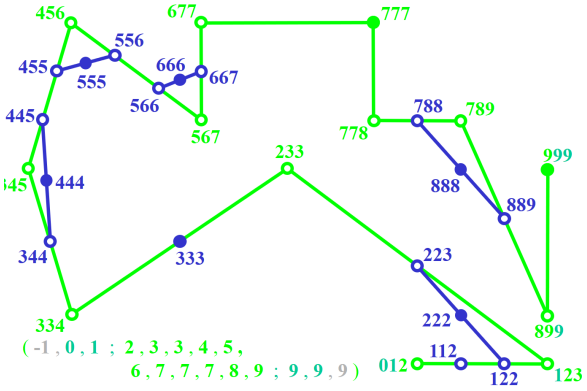
krijgen, zodat het punt 223 mooi in het midden van het lijnstuk $[P[1,2,3]P[2,3,3]]$ ligt. Een leuke eigenschap is, dat als de knopenvector geen gaten vertoond, dat het lijnstuk altijd mooi opgesplitst wordt. Als je op een lijnstuk dan 2 hulppunten moet introduceren, zal het eerste punt op 1/3 van het lijnstuk liggen, en het tweede punt op 2/3. Als je 3 hulppunten moet introduceren, zal het eerste punt op 1/4 van het lijnstuk liggen, enz... Een voorbeeld van een knopenvector met gaten, waardoor je dus best manueel uitrekend, is $(0, 1, 3; 4, 6; 7, 9, 10)$. Alle nieuwe hulppunten kunnen zo berekend worden, en is te zien op figuur 3.10.

Punten waarvan slechts 1 argument van de blossomnotatie verschillen, en die op een ander lijnstuk liggen, kunnen geïnterpoleerd worden. Teken een lijnstuk door elk paar punten waarvoor dit van toepassing is. Op figuur 3.11 wordt dit uitgewerkt, samen met het introduceren van de nieuwe hulppunten.

Tot slot moeten nog enkel de punten verbonden worden waarvan de blossomnotatie bestaat uit een herhaling van slechts één argument. In dit voorbeeld zijn dat de punten 222,



Figuur 3.10: Fase 3 bézier representatie van een polynomiale NURBS.



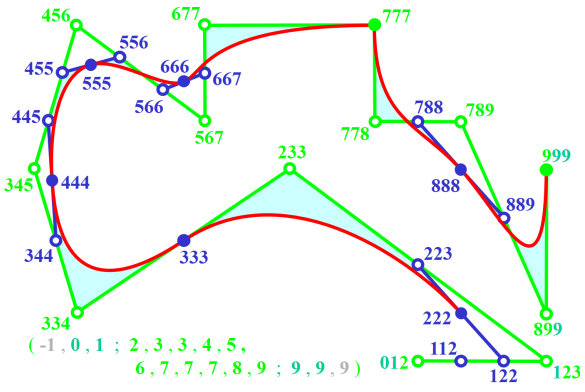
Figuur 3.11: Fase 4 bézier representatie van een polynomiale NURBS.

333, 444, 555, 666, 777, 888 en 999. Zoek eerst in de knopenvector op of er knooppunten zijn met een hogere multipliciteit. De continuïteit in een knooppunt horend bij een knoop met multipliciteit μ wordt gegeven door $C^{k-1-\mu}$.

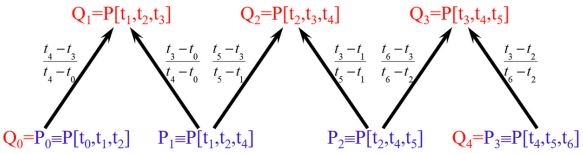
Het knooppunt 3 heeft een multipliciteit van 2, zodat slechts $C^{4-1-2} = C^1$ continuïteit voldaan is. Het knooppunt 7 heeft een multipliciteit van 3, zodat daar slechts C^0 voldaan is. Dit is belangrijk bij het tekenen van de kromme, zoals te zien op figuur 3.12.

- (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)
  - Voeg het nieuwe punt zoveel keer toe als de multipliciteit van de knoop aangeeft in de knopenvector. Daarna moeten er zoveel iteratatie van het algoritme van de Boor uitgevoerd worden als de multipliciteit van de nieuwe knoop. Figuur 3.13, 3.14 en 3.15 geven respectievelijk het toevoegen van een knoop met multipliciteit 1, 2 of 3 weer.
- (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)
  - Als de toe te voegen virtuele knoop de meest extreme virtuele knoop wordt, dan vervangen we de meest extreme virtuele knoop met deze nieuwe knoop en blijven alle andere controlepunten ongewijzigd aangezien geen enkel van de controlepunten afhankelijk is van de meest extreme virtuele knoop.

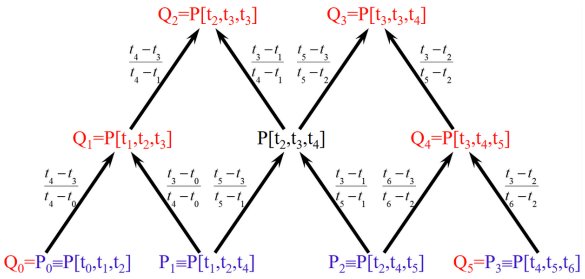
$$(t_0, t_1, t_2, ..., t_{k-2}; t_{k-1}, t_k, t_{k+1}, ...) \rightarrow (t_x, t_1, t_2, ..., t_{k-2}; t_{k-1}, t_k, t_{k+1}, ...)$$



Figuur 3.12: Fase 5 bézier representatie van een polynomiale NURBS.



Figuur 3.13: Knoop toevoegen met multiplicteit 1 bij een NURBS van graad 3.



Figuur 3.14: Knoop toevoegen met multiplicteit 2 NURBS van graad 3.

- In het andere geval moeten de virtuele knopen opgeschoven worden tot dat de virtuele knoop op zijn plaats zit:

  $$(t_0, t_1, t_2, ..., t_j, t_{j+1}, ..., t_{k-2}; t_{k-1}, t_k, t_{k+1}, ...) \rightarrow (t_1, t_2, ..., t_j, t_x, t_{j+1}, ..., t_{k-2}; t_{k-1}, t_k, t_{k+1}, ...)$$

  Hierbij worden er j virtuele knopen opgeschoven, en moeten ook j controlepunten vervangen worden. Voer hiervoor het algoritme van de Boor op, met als basis van het schema de ??? _ToDo: niet gevonden

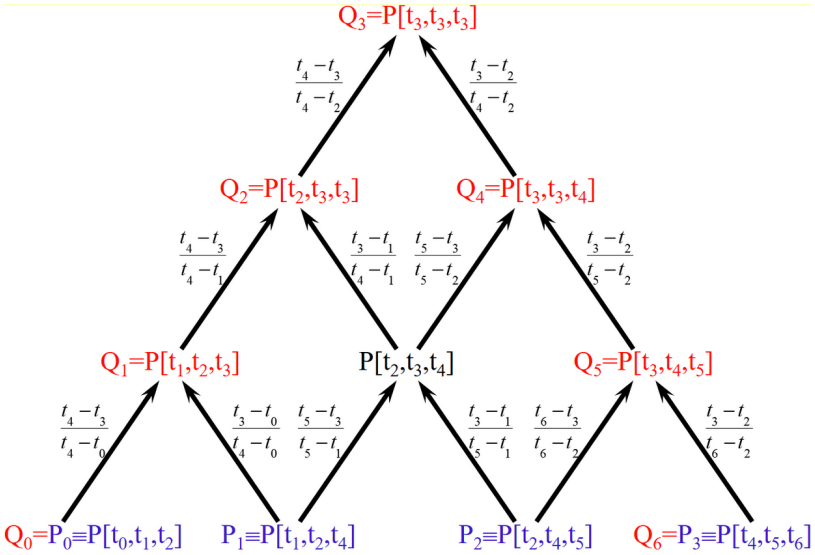
- (11) berekening en constructie van de controlepunten van de *open-uniforme* representatie van een (polynomiale) NURBS met een *uniforme* knopenvector

  - Bij een uniforme knopenvector vormen alle knopen een rekenkundige rij. Bij een open-uniforme knopenvector vormen enkel de reële knopen een rekenkundige rij. De meest extreme reële knopen worden k maal herhaald. Een open-uniforme NURBS begint en eindigt op respectievelijk het eerste en laatste datapunt.

    Zo wordt de uniforme knopenvector $\{-1, 0, 1; 2, 3, 4, 5; 6, 7, 8\}$ in open-uniforme vorm: $\{2, 2, 2; 2, 3, 4, 5; 5, 5, 5\}$. Maak gebruik van de interpolatieblokken om de nieuwe controlepunten te achterhalen.

- (12,19) berekening en constructie van de controlepunten van de *uniforme* representatie van een polynomiale of rationale NURBS met een *open-uniforme* knopenvector

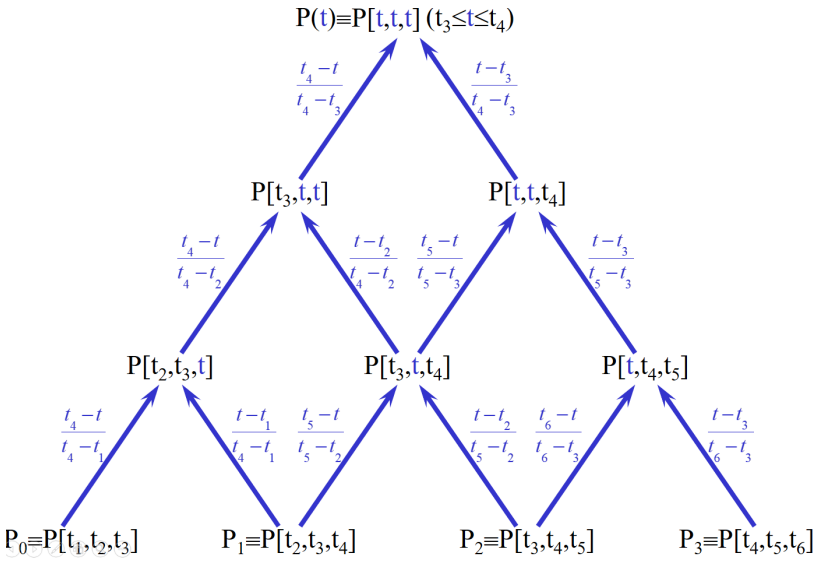
  - Bij de uniforme knopenvector vormen alle knopen een rekenkundige rij. Bij een open-uniforme knopenvector vormen enkel de reële knopen een rekenkundige rij. De meest extreme reële knopen worden k maal herhaald. Een open-uniforme NURBS begint en eindigt op respectievelijk het eerste en laatste datapunt.



Figuur 3.15: Knoop toevoegen met multiplicteit 3 NURBS van graad 3.

Zo wordt de open-uniforme knopenvector $\{2,2,2;2,3,4,5;5,5,5\}$ in uniforme vorm: $\{-1,0,1;2,3,4,5;6,7,8\}$ Maak gebruik van de interpolatieblokken om de nieuwe controlepunten te achterhalen.

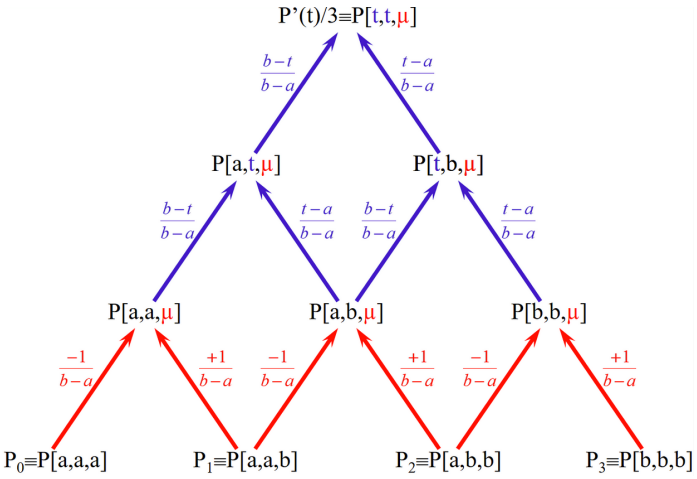
- (12) de Boor constructie (van een punt met specifieke parameterwaarde) van een (polynomiale) NURBS
  - Hierbij wordt een piramidaal schema gebruikt dat identiek is aan het schema dat gebruikt wordt voor het algoritme van de Casteljau. Het verschil is dat bij de Boor de inputgegevens bestaan uit louter controlepunten, zonder datapunten. De constructie is te zien op figuur 3.16.



Figuur 3.16: de Boor constructie.

- (13) constructie van de *hodograaf* van een Bézier kromme of spline

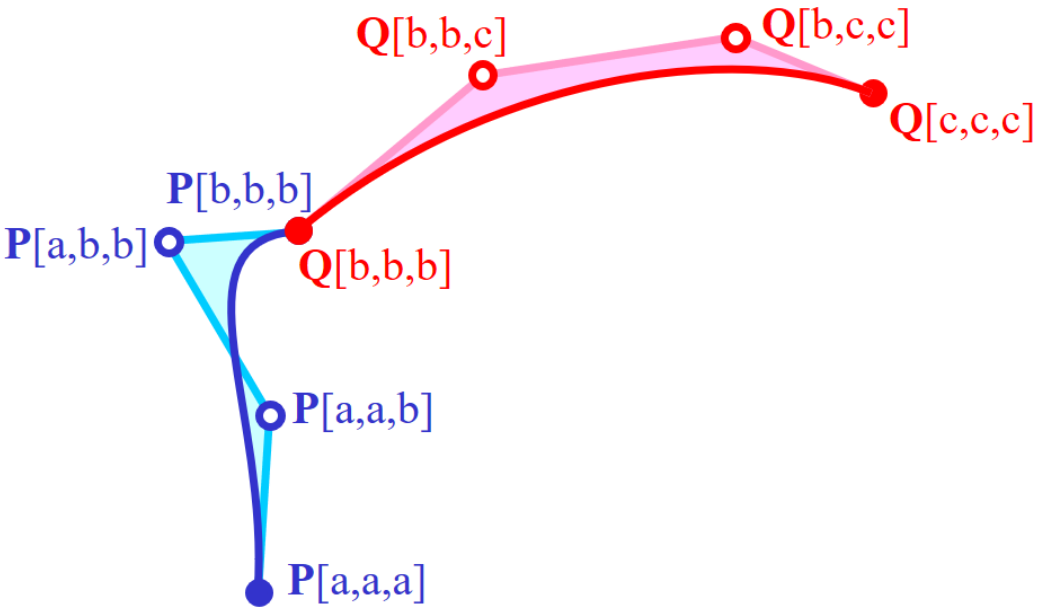
- Stel het schema van de Casteljau voor de gegeven Bézierkromme, of het schema van de Boor voor de gegeven spline op, maar voer in de eerste stap de blokken van het afgeleide type in, zoals weergegeven op figuur 3.17. De resulterende hodograaf is zelf een Bézierkrommme/spline van een graad minder en wordt gegeven door de resterende blokken van het interpolerende type.



Figuur 3.17: Constructie van het de Casteljau schema voor een hodograaf.

- (13-16) vaststellen van de continuïteit in de knooppunten van Bézier splines (*stelling van Stärk*)
  - **Stelling van Stärk:** Men bereikt C^k continuïteit indien de blossomargumenten voor elk van de $2k$ naburige controlepunten hetzelfde punt in de ruimte oplevert, of men nu de argumenten evalueert in de blossompolynoom van het ene segment, of in de blossompolynoom van het andere segment.
  - Veronderstel een linkersegment met data- en controlepunten $P[a,a,a], P[a,a,b], P[a,b,b]$ en $P[b,b,b]$, en een rechtersegment, bepaald door $Q[b,b,b], Q[b,b,c], Q[b,c,c]$ en $Q[c,c,c]$ met gemeenschappelijk punt $P[b,b,b] \equiv Q[b,b,b]$. Dit is ook te zien op figuur 3.18.
    Aangezien we dit gemeenschappelijk punt hebben is C^0 continuïteit voldaan. Om C^1 continuïteit te bereiken moet, indien men de argumenten van controlepunt $P[a,b,b]$ van het linkersegment evalueert in de blossom van het rechtersegment, dit een punt $Q[a,b,b]$ opleveren dat samenvalt met $P[a,b,b]$. Hetzelfde moet gelden tussen controlepunt $Q[b,b,c]$ van het rechtersegment, en het punt $P[b,b,c]$ van het linkersegment, bekomen door de argumenten te evalueren in de blossom van het linkersegment. Om C^2 continuïteit te hebben doet men hetzelfde voor punten $P[a,a,b]$ en $Q[b,c,c]$. Om C^3 continuïteit te hebben doet men hetzelfde voor punten $P[a,a,a]$ en $Q[c,c,c]$. Dit is te zien op figuur 3.19.

- (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).
  - x
  - De matrix voor een Bézierkromme van willekeurige graad n kan als volgt opgesteld worden:
    1. Construeer de driehoek van Pascal, in de linkerbovenhoek van een $(n+1)\cdot(n+1)$ matrix, vertrekkend van de rechterbovenhoek. De overige coëfficiënten blijven nul.
    2. Vermenigvuldig alle coëfficiënten van kolom j, met de coëfficiënt in rij j van de eerste kolom.

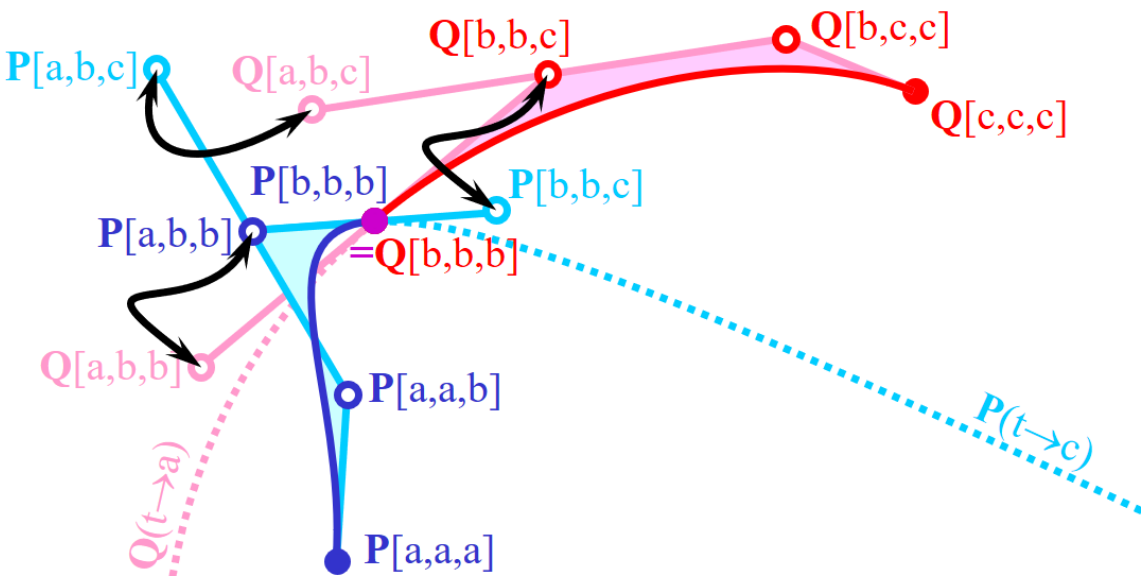


Figuur 3.18: Een Bézier spline.

3. Keer het teken om van alle coëfficiënten op de even nevendiagonalen.

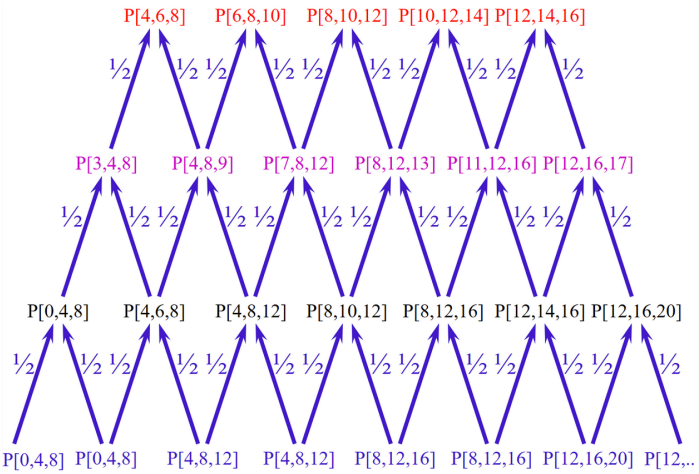
Toegepast op de figuur ($n = 3$ aangezien 4 datapunten):

$$\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$



Figuur 3.19: De stelling van Stärk.

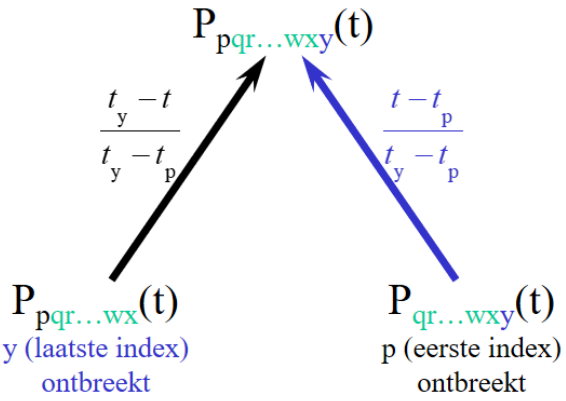
- (18,19) constructie van een *benadering door lijnstukken* van een uniforme NURBS door toepassing van het algoritme van Lane & Riesenfeld
  - Dit is een eenvoudig iteratief proces: pas altijd de helft af tussen twee punten, zoals te zien op figuur 3.20.



Figuur 3.20: Het algoritme van Lane en Riesenfeld.

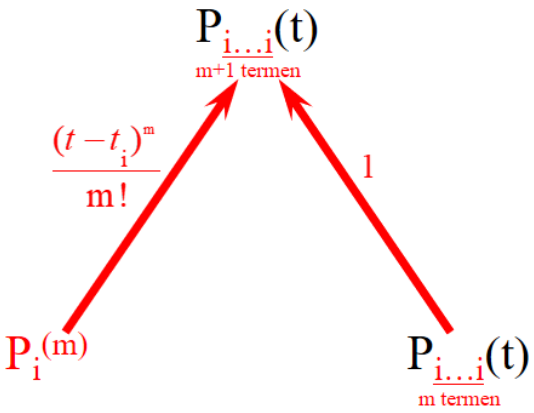
- (20) constructie van een *triangulair schema* met behulp van het veralgemeend algoritme van Neville (voor een specifieke configuratie van inputgegevens), en berekening hieruit van de *gewichtsfuncties* en de *matrixrepresentatie*
  - Het veralgmeend algoritme van Neville laat toe om de rekenregels van Hermite interpolatie op te stellen, aangezien Hermite interpolatie ook nog gebruik maakt van richtingsvectoren en eventueel krommingen, die gedefinieerd worden op hetzelfde punt, zodat het punt een hogere multipliciteit krijgt. Hermite interpolatie gebruikt slechts twee datapunten P_0 en P_1, waarvoor men meestal respectievelijk $t = 0$ en $t = 1$ neemt. Bij het veralgemeend algoritme van Neville zijn er twee mogelijkheden:

1. Indien er minstens twee verschillende indices in $P_{pqr\ldots wxy}(t)$ optreden, dan blijft de traditionele versie van het algoritme onveranderd geldig, zoals te zien op figuur 3.21.



Figuur 3.21: Lineaire interpolatie met het algoritme van Neville.

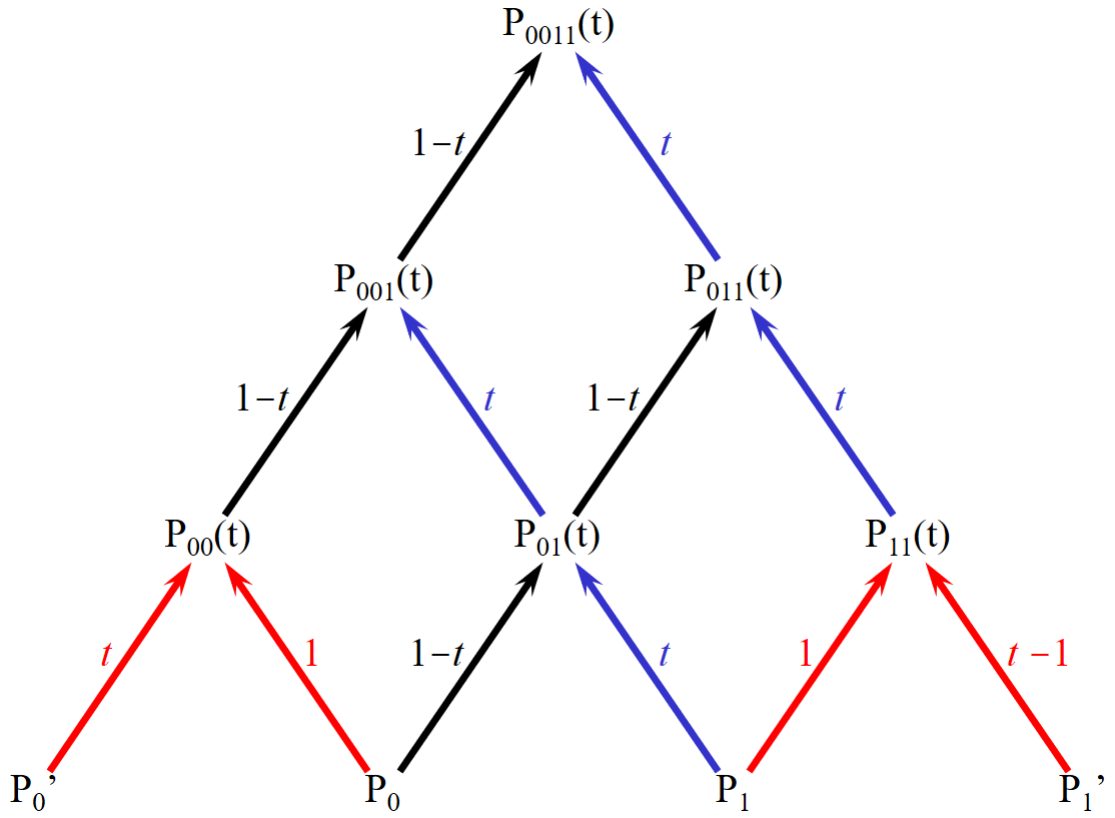
2. Indien echter in de labeling van een hulppunt, of van het eindresultaat, slechts één enkele index i optreedt, met multipliciteit $\mu_i = m + 1$, dan moet op het lager niveau slechts één enkel hulppunt geïntroduceerd worden: het hulppunt waarin de index i optreedt met multipliciteit $\mu_i = m$. Ter vervanging van lineaire interpolatie moet nu gebruik gemaakt worden van Taylor reeksontwikkelingen, zoals te zien op figuur 3.22.



Figuur 3.22: Taylor reeksontwikkeling met het algoritme van Neville.

Veronderstel nu een Hermite interpolatie van graad drie. Dit heeft twee datapunten P_0 en P_1 en de richtingsvectoren in deze punten P_0' en P_1'. Punt $P_{0011}(t)$ is de top van het piramidaal schema. Op figuur 3.23 wordt dit schema voorgesteld.
De gewichtsfuncties kunnen berekent worden via de volgende stappen:

1. Keer elke pijl van het schema om.
2. Plaats de constante 1 op de top van de piramide.
3. De mengfunctie horend bij een specifiek basisdatapunt P_i, kan bekomen worden door in het aangepaste diagram achtereenvolgens:
   (a) het aantal alternatieve paden te zoeken die tot dit punt leiden (eenvoudig via de driehoek van Pascal),



Figuur 3.23: Constructie van het triangulair schema.

(b) de gewichtsfactoren langs een specifiek pad te vermenigvuldigen,
(c) de bijdrage van elk individueel pad te sommeren.

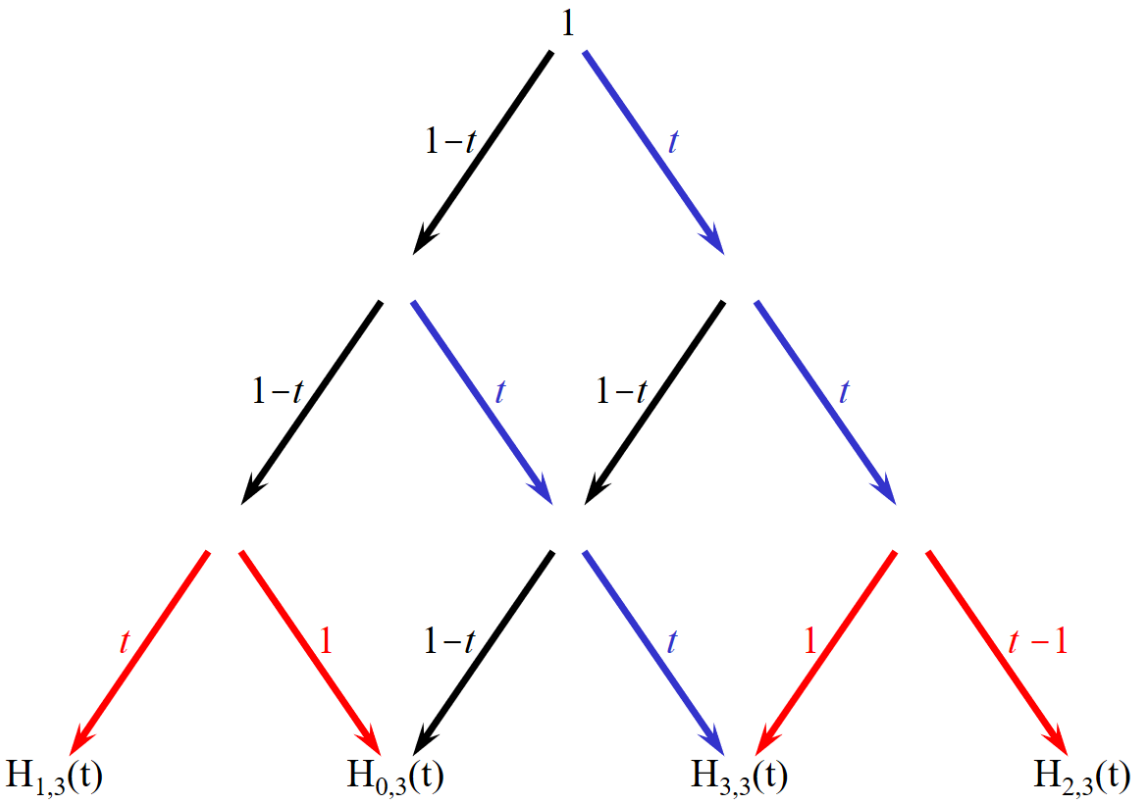
Op figuur 3.24 wordt het omgekeerde schema getoond.
De mengfuncties worden dan:

$$H_{1,3}(t) = \Big[t(1-t)^2\Big] = t^3 - 2t^2 + t$$
$$H_{0,3}(t) = \Big[1(1-t)^2 + 2(1-t)^2 t\Big] = 2t^3 - 3t^2 + 1$$
$$H_{3,3}(t) = \Big[t^2 + 2t^2(1-t)\Big] = -2t^3 + 3t^2$$
$$H_{2,3}(t) = \Big[t^2(t-1)\Big] = t^3 - t^2$$

Omgevormd tot de matrixrepresentatie:

$$P(t) = (t^3, t^2, t, 1) \cdot \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} P_0 \\ P_1 \\ P_0' \\ P_1' \end{pmatrix}$$

CONTROLE: Som van de waarden van de rijen voor elk datapunt moet 0 zijn: $-24 - 12 - \frac{1}{2} + \frac{1}{2} + a_4 + a_5 = 0$.

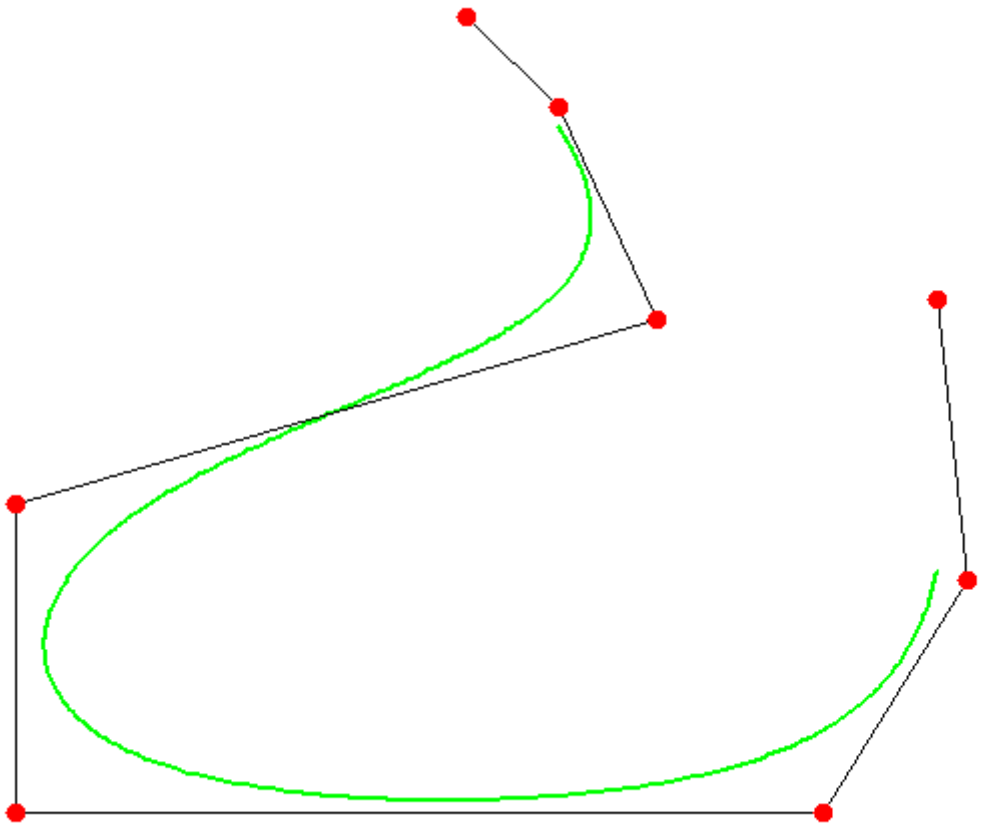


Figuur 3.24: Het inverse triangulair schema dat op figuur 3.23 geconstrueerd werd.

Hoofdstuk 4

Oefening 17 januari 2019

1. Geef de blossomnotatie van de controlepunten van de open-uniforme B-spline met graad vier (orde vijf) op figuur 4.1. Geef ook de knopenvector.
2. Geef de blossomnotatie van de controlepunten met een uniforme knopenvector die dezelfde B-spline representeerd. Geef ook de knopenvector.
3. Construeer de nieuwe controlepunten van de uniforme representatie.
4. Hoeveel segmenten bevat de oorspronkelijke spline? Indien het aantal segmenten verdubbeld wordt, hoeveel controlepunten zijn er dan nodig?
5. Construeer deze controlepunten door het algoritme van Lane en Riesenfeld toe te passen.



Figuur 4.1