

# Ruimtemeetkunde op de computer

L.M. Doorman/H.B. Verhage

OW & OC, RU Utrecht

## Samenvatting

Kan de computer bij de ruimtemeetkunde ondersteuning bieden? Het ruimtemeetkundeprogramma van de HAWEX lijkt hiervoor wel geschikt. We bespreken twee computerprogramma's die ontwikkeld zijn ten behoeve van de HAWEX. We vergelijken ze met elkaar en bekijken hoe ze ondersteuning kunnen bieden bij deze nieuwe ruimtemeetkunde.

## Inleiding

De ruimtemeetkunde van het havo-bovenbouwprogramma krijgt bij het Hawex-project een geheel ander aanzien. In plaats van de huidige vectormeetkunde wordt er veel meer aandacht geschonken aan de ontwikkeling van ruimtelijk inzicht. Dit wordt onder meer gerealiseerd door andere definities voor ruimtelijke objecten zoals de balk, de kubus en het prisma [1]. In plaats van een definitie als:

'Een prisma is een lichaam begrensd door enige vlakken, die elkaar volgens evenwijdige lijnen snijden, en twee evenwijdige vlakken.'

werd er gekozen voor het volgende recept:

'Een prisma ontstaat door een veelhoek (grondvlak) evenwijdig aan de beginstand langs een lijn te verschuiven.'

In het artikel *Dynamiek in de ruimte* maakt Henk van der Kooij duidelijk waarom het tweede recept de voorkeur geniet boven de eerste statische definitie: 'Zo'n lichaam zie je ontstaan op het moment dat je een veelhoek hebt gekozen en een schuifrichting hebt bepaald.' Het gebruik van deze dynamiek heeft een belangrijke positie in het nieuwe ruimtemeetkundeprogramma. Het probleem is dat de opbouw van een figuur volgens bovenstaand recept moeilijk met de hand na te doen is. Voor het visualiseren van zo'n zogenaamde dynamische opbouw is een computer natuurlijk heel goed bruikbaar.

In dit artikel gaan we verder in op het gebruik van twee programma's bij de ruimtemeetkunde, een menugestuurd programma *RuimFig* en een Logo-pro-

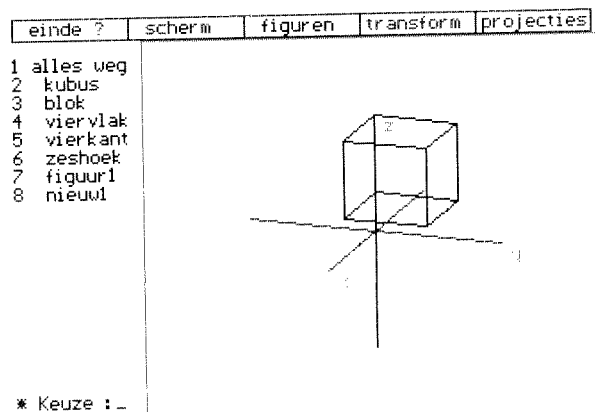
gramma *Pakket3D*. We zien deze programma's niet als vervanging van de gebruikelijke relikwieën zoals draadmodellen, maar juist als een extra hulpmiddel naast deze voorwerpen.

## RuimFig

RuimFig is een menugestuurd programma. Dat wil zeggen dat alle mogelijkheden van het programma zijn ondergebracht in menu's en dat de computer de opdrachten uitvoert die gekozen zijn uit de menu's. De indeling is als volgt: (zie figuur 1.)

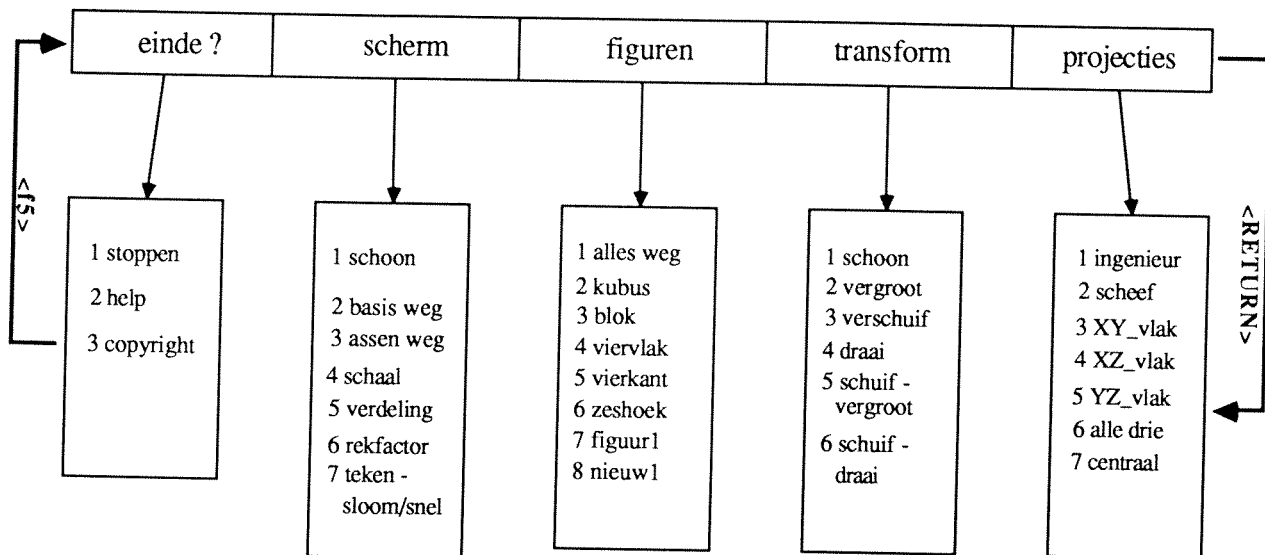
Omwille van de overzichtelijkheid worden de verschillende opdrachten die een programma kan uitvoeren vaak ondergebracht in submenu's, die hier elk maximaal zes à acht menukeuzes bevatten. RuimFig is geschreven in TurboPascal.

Na het starten kom je terecht in het menu figuren:



Startscherm met kubus

fig. 2



Menu's in RuimFig

fig. 1

In de bovenste balk (zie figuur 2) staan de menu's vermeld, dit heet de *menubalk*. Links staat de *menukaart*, daar worden de verschillende mogelijkheden binnen het huidige menu vermeld. Onder de menukaart vindt de communicatie met het programma plaats. Er is nog enige ruimte voor het geval de computer meer informatie nodig heeft (bijvoorbeeld bij transformaties). Omkaderd is het *grafische scherm*, daarbinnen wordt alles getekend. Een keuze maak je door het betreffende nummer te typen gevolgd door <return>.

### De eerste opgaven

Met RuimFig is op een aantal scholen geëxperimenteerd. De eerste paar opgaven uit de werkbladen gaan over een kubus. De leerling wordt gevraagd met behulp van de verschillende aanzichten de positie van de kubus te bepalen (zie figuur 2). Uit observaties bleek dat dit veel moeite kostte. Onder andere wordt dat veroorzaakt doordat de computer de figuur altijd over de assen heen tekent. Vervolgens wordt gevraagd of ze door het verschuiven van de kubus het volgende plaatje kunnen maken:

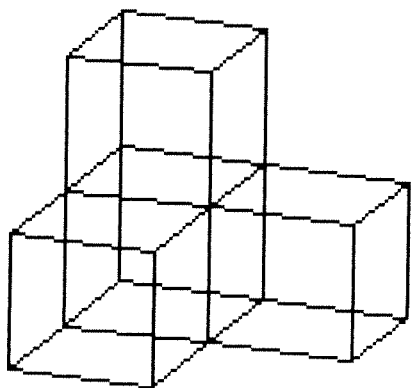


fig. 3

De meeste leerlingen weten dan wel een eerste aangrenzende kubus op het scherm te laten verschijnen, bijvoorbeeld door de eerste kubus te verschuiven met

translatie  $[2\ 0\ 0]$  (de lengte van een ribbe 2). Dan wordt het moeilijk. Vaak werd er dan verschoven over  $[0\ 0\ 2]$  en ontstond er dit:

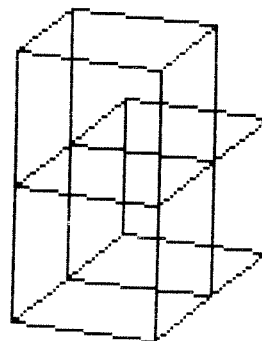


fig. 4

De moeilijkheid schuilt hem in het feit dat de *laatst getekende kubus* de kubus is die verschoven gaat worden. Inmiddels gegrepen door de kubus wordt er weer opnieuw geprobeerd. Eerst weer over  $[2\ 0\ 0]$  verschuiven, daarna  $[-2\ 0\ 0]$ , en dan  $[0\ 0\ 2]$ .

Als dan de laatste kubus op het scherm moet komen wordt er weer eerst teruggeschoven en als laatste de opdracht gegeven, verschuif over  $[0\ 2\ 0]$ . Dat kan ook sneller. Deze laatste twee translaties kunnen natuurlijk ook in één translatie verwerkt worden. Door de kubus over  $[0\ 0\ -2]$  te verschuiven wordt in één keer het goede eindresultaat bereikt.

Het bedieningsgemak van menugestuurde programma's is groot, tenminste als de verschillende menukeuzes op overzichtelijke wijze zijn ingedeeld in de submenu's.

Een nadeel van menugestuurde programma's kan zijn dat ze vrij rigide zijn. De ontwerper/programmeur heeft bepaald wat mogelijk is met het programma, en daar kan verder niets meer aan veranderd worden.

Bij RuimFig bijvoorbeeld, is in het programma vastgelegd dat verschuiven altijd werkt op de laatst geteken-

de figuur. Uit het hiervoor geschetste gedrag van de leerlingen blijkt, dat ze eigenlijk wilden schuiven met de oorspronkelijke figuur, maar dat laat het programma niet toe.

### Pakket 3D

Er is ook een programma in Logo ontwikkeld, vergelijkbaar met RuimFig, dat leerlingen meer vrijheid geeft om zelf aan de computer op te geven wat er moet gebeuren. Alvorens te deze vrijheid optimaal kunnen benutten, moet er echter wel wat gebeuren: ze moeten zich het voor dit doel ontwikkelde Logo-vocabulaire eigen maken. Bij de voorbeelden in dit artikel maken we gebruik van de LCN-Logo-versie van dit programma [2].

Met Logo kan heel wat meer dan de bekende 'teken huisjes, teken veelhoek'-activiteiten die met jonge kinderen worden gedaan. Logo is een taal die gemakkelijk is uit te breiden met nieuwe woorden, zodat voor elk gewenst type toepassing een aantal woorden aan de taal toegevoegd kan worden. De primitieven van de taal én de eerder gedefinieerde woorden vormen de bouwstenen voor nieuwe woorden. Bij LCN-Logo speelt het uitbreiden van de taal zich op dezelfde plek af als het evalueren van Logo-woorden: namelijk in de editor. In principe is er geen onderscheid tussen programmeur en gebruiker, beiden werken op hetzelfde niveau.

Het Logo-programma Pakket3D is niet meer dan een prototype, door middel waarvan enige ideeën concreet zijn uitgewerkt. Het programma is op beperkte schaal beproefd met leerlingen en is verder opgenomen in de NIVO-nascholingscursus Computers en Wiskunde. Het programma bestaat in feite uit een uitbreiding van de taal Logo met een aantal woorden, gericht op het tekenen van en manipuleren met ruimtelijke figuren in een assenstelsel. Centraal staat de opdracht *teken*, die een object op het scherm tekent. Een object is een (ruimtelijke) figuur die is vastgelegd door de coördinaten van de hoekpunten en de verbindingen daartussen. Om een indruk te geven van het type opdrachten dat mogelijk is, noemen we er een paar:

```
teken <object>
teken vergroot <object>
    <faktor>
teken verschuif <object>
    <vektor>
```

*Vergroot* en *verschuif* zijn functies, die een nieuw object opleveren. Dat nieuwe object wordt door *teken* op het beeldscherm getekend. De functies die een bewerking op het object uitvoeren, kunnen ook met elkaar gecombineerd worden, bijvoorbeeld:

```
teken verschuif vergroot <object>
    <faktor>
    <vektor>
```

Om met zo'n 'woordenlijst' uit de voeten te kunnen, is het nodig iets te weten van de syntactische regels die vastleggen hóe van de toegestane woorden zinnen gemaakt kunnen worden die LCN-Logo verstaat. Het valt niet te ontkennen dat er wat tijd voor nodig is om in zo'n 'taaltje' thuis te raken. Daar staat in dit geval tegenover, dat het werken met Pakket3D uiteindelijk meer mogelijkheden en vrijheden geeft dan het wandelen door de menu's van RuimFig.

### Kubusjes tekenen in LCN-LOGO

De eerder genoemde opgave over het tekenen van de vier kubusjes gaat in LCN-Logo door het intypen en evalueren van de volgende 'zinnen':

<pre>maak "kubus     verschuif kubus1         "[1 1 1]  teken kubus  teken verschuif kubus     "[2 0 0]  teken verschuif kubus     "[0 2 0]  teken verschuif kubus     "[0 0 2]</pre>	<p>Toelichting:</p> <p><i>kubus1</i> is een eerder gedefinieerd object, n.l. een kubus met hoekpunten <math>(\pm 1, \pm 1, \pm 1)</math>. <i>kubus</i> ontstaat door <i>kubus1</i> te verschuiven over de vector <math>[1 1 1]</math></p> <p><i>kubus</i> wordt getekend</p> <p><i>kubus</i> wordt verschoven over <math>[2 0 0]</math> en getekend</p> <p>idem, nu over <math>[0 2 0]</math></p> <p>idem, nu over <math>[0 0 2]</math></p>
---	---

Het uitgangspunt bij deze oplossing voor de kubusstapel is het object *kubus*, dat één keer gedefinieerd wordt en vervolgens niet meer verandert. Een nieuwe *kubus* wordt steeds verkregen door de oorspronkelijke te verschuiven.

Een oplossing analoog aan de aanpak die in RuimFig gevolgd wordt, waarbij 'kubus' steeds gelijk gemaakt wordt aan het laatst getekende object, ligt in dit geval bepaald niet voor de hand. Het is mogelijk door de verschoven *kubus* niet meteen te tekenen, maar eerst een naam te geven waardoor de nieuwe *kubus* als object beschikbaar blijft:

<pre>maak "kubus     verschuif kubus         "[2 0 0]  teken kubus</pre>	<p>Toelichting:</p> <p>Uit de oude <i>kubus</i> wordt een nieuwe <i>kubus</i> gemaakt door die te verschuiven.</p> <p>De nieuwe <i>kubus</i> wordt getekend.</p>
--	--

De volgende opdrachten laten zich raden:

```
maak "kubus
    verschuif kubus
        "[-2 2 0]

teken kubus

maak "kubus
    verschuif kubus
        "[0 -2 2]

teken kubus
```

## Computer en dynamiek

In de inleiding van dit artikel is al opgemerkt dat de dynamische opbouw van figuren een belangrijk element is uit de Hawex-meetkunde. De twee programma's kunnen voor het zichtbaar maken van deze dynamiek goed gebruikt worden. Laten we bijvoorbeeld het recept van een zeszijdig prisma bekijken:

*Een zeszijdig prisma ontstaat door een zeshoek evenwijdig aan de beginstand langs een lijn te verschuiven.*

Beide programma's hebben een zeshoek als standaardfiguur opgeslagen.

Bij RuimFig wordt het dan:

- > Kies in het menu FIGUREN de zeshoek als basisfiguur.
- > Ga naar het menu TRANSFORM.
- > Kies de opdracht 'verschuif'.
- > Voer de translatie [0 0.2 0.3] in met aantal 5, dan wordt de volgende figuur in het tekenveld opgebouwd:



fig. 5

Bij Pakket3D volstaat slechts één opdracht om deze tekening te maken:

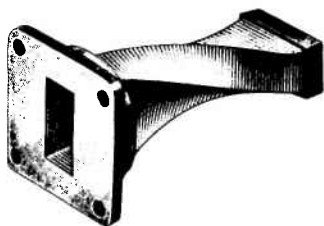
```
schuifserie zeshoek
"[0 0.2 0.3]
5.
```

Uit de beschrijvingswijze van deze opgave blijkt nog eens het typerende verschil tussen de twee programma's. Activiteiten met RuimFig kunnen haast niet anders beschreven worden dan in termen van te verrichten handelingen binnen de menustructuur: 'Kies uit het menu ...'

De activiteiten met Pakket3D laten zich beschrijven door rechtstreeks te noteren wat het programma moet doen: 'teken ...' Je zou kunnen spreken van respectievelijk *indirecte* en *directe conversatie* met de computer.

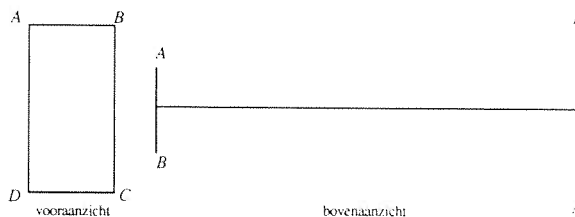
Een ingewikkelder opgave in dit verband is de volgende som uit het Hawex-pakket *Verkenning in de Ruimte*:

Door een rechthoek langs een lijn te verschuiven en daarbij gelijkmatig te roteren, kan het middenstuk van dit object ontstaan. Dit vraagstuk gaat alleen over dit middenstuk.



ELECTRONIC  
WAVE GUIDE

- >a Hieronder zie je de rechthoek in de beginstand. Teken in het vooraanzicht op het werkblad de rechthoek in de eindstand. Teken ook de 25e rechthoek (neem aan dat er 75 rechthoeken zijn).



- >b Past dit lichaam precies in een cilinder?
- >c Teken in het bovenaanzicht op het werkblad de baan van A.
- >d Het is mogelijk door het lichaam heen te kijken, maar de opening is natuurlijk niet rechthoekig. Teken die opening.
- >e De afstand tussen de eerste en de laatste rechthoek is 10. Probeer de inhoud van het lichaam te berekenen. Verklaar je methode.

fig. 6

RuimFig kan als volgt bij deze opgave gebruikt worden. Maak een nieuwe basisfiguur met hoekpunten (0,2,1), (0,2,-1), (0,-2,-1), (0,-2,1). Er ontstaat na de opdracht schuif-draai (met de parameters schuif: [-0.3 0 0]; draai-as: [1 0 0]; hoek: 5 (in graden); aantal 18) de volgende figuur:

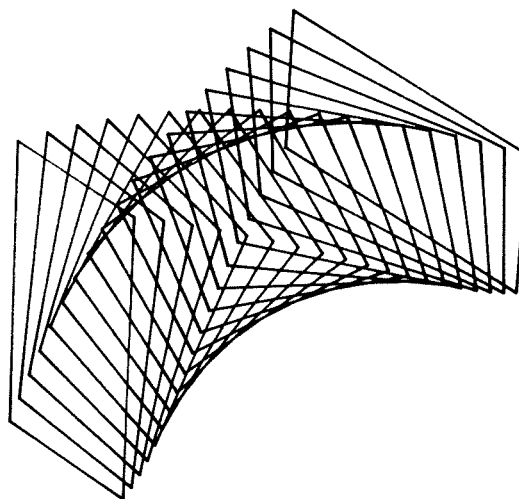


fig. 7

Het zijaanzicht en het vooraanzicht geeft:

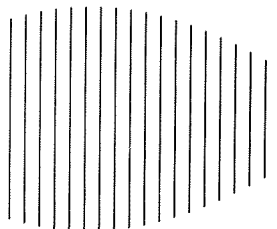


fig. 8a

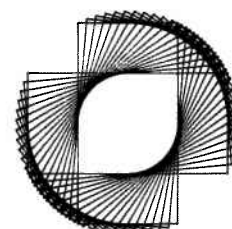


fig. 8b

Ook is het aardig om dezelfde opdrachten te geven met als basisfiguur slechts de vier hoekpunten. De bovenstaande twee aanzichten worden dan: (zie figuur 9a en 9b):

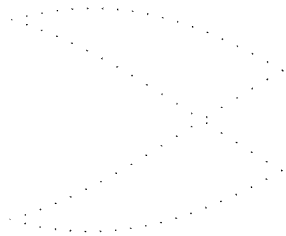


fig. 9a

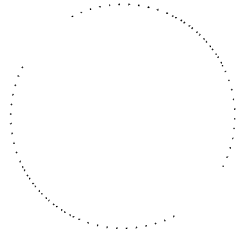


fig. 9b

De opdracht *schuifdraai* kwam oorspronkelijk niet voor in het Logo-programma, maar een toevoeging op basis van de woorden die er voor schuiven en draaien al zijn, is gauw gemaakt. De aanroep van de opdracht *schuifdraai* moet worden:

```
schuifdraaix <objekt>
             <vektor>
             <hoek>
             <aantal>
```

Om het niet al te ingewikkeld te maken beperken we ons hier tot een verschuiving gecombineerd met een draaiing om de *x*-as, in plaats van een vrij te kiezen as. Het draaien kost in Logo namelijk helaas erg veel tijd, waardoor de opdracht voor het algemene geval erg traag zou worden.

Evaluatie van:

```
schuifdraaix rechthoek
             "[ -0.3 0 0 ]
             5
             18
```

moet dan figuur 7 geven.

Om het vooraanzicht te krijgen moet deze opdracht vooraf gegaan worden door:

```
projekteer achtervlak
```

en voor het zijaanzicht door:

```
projekteer zijvlak
```

De woorden *projekteer*, *achtervlak* en *zijvlak* maken ook deel uit van Pakket3D. Hun betekenis spreekt voor zich.

Een definitie van de opdracht *schuifdraaix* die inderdaad het gewenste resultaat oplevert, is:

```
MAAK schuifdraaix
      FUNK MET objekt
              richting
              hoekstap
              aantal
      LOKAAL vektor
              hoek
      teken objekt
      MAAK vektor
              "[ 0 0 0 ]
      MAAK hoek
              0
      HERHAAL aantal
              teken verschuif draaix objekt
```

```
MAAK hoek
      plus hoek
      hoekstap
MAAK vektor
      vektorsom vektor
      richting
```

Op deze manier kan ook de gebruiker zelf nieuwe woorden maken en aan de taal toevoegen. Belangrijke ingrediënten van deze definitie zijn de eerder gemaakte functies *verschuif* en *draaix*. De definitie van *schuifdraaix* geeft een indruk van de syntax van LCN-Logo. Zo zijn de woorden in hoofdletters bijvoorbeeld taalprimitieven. Verder valt de consequente keuze voor de prefix-notatie op (eerst wordt de operator genoteerd, daarna de argumenten). De notatie voor  $a+b*c$  is:

```
plus a
      maal b
      c
```

Bepaald even wennen! Juist op elementair niveau sluit de notatiewijze slecht aan bij wat gebruikelijk is in de wiskunde. Alhoewel er in principe natuurlijk veel voor een consequente notatiewijze te zeggen is, kan dit niet anders dan een ernstig nadeel van LCN-Logo genoemd worden.

## Projecties

Naast het gebruik van de computer voor het zichtbaar maken van deze dynamische processen is er nog een ander punt waar de machine functioneel kan zijn. Beide programma's hebben namelijk de mogelijkheid om figuren in een aantal projecties te tekenen. Projecties kunnen vaak erg verwarrend zijn. Dit is bijvoorbeeld het geval bij een opgave uit de werkbladen van RuimFig over een kunstwerk in de buurt van Eindhoven. Op het scherm werd het volgende figuur zichtbaar (met hoekpunten  $(-2,-2,0)$ ,  $(2,2,0)$ ,  $(-2,2,4)$  en  $(2,-2,4)$ ):

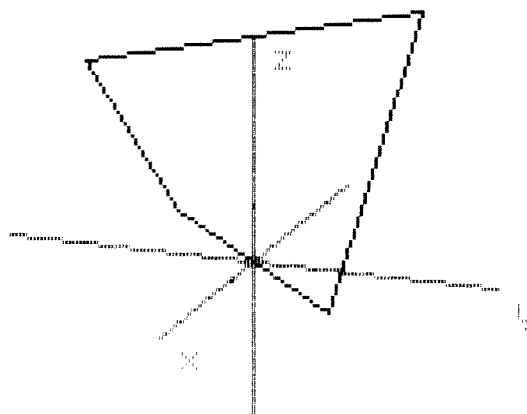


fig. 10

Ook het zij-, boven- en vooraanzicht brachten niet altijd opheldering omtrent de vorm van het kunstwerk. Met de foto's ernaast is het eenvoudiger.

Over dit kunstwerk werden een aantal vragen gesteld, onder andere of het zó kon draaien dat het zijaanzicht de vorm van een driehoek had. (Rond welke as? Over

hoeveel graden?) Door het gebruik van foto's bij deze opgave kwamen we op nieuwe ideeën. Deze komen in de volgende paragraaf aan de orde.



fig. 11a

fig. 11b

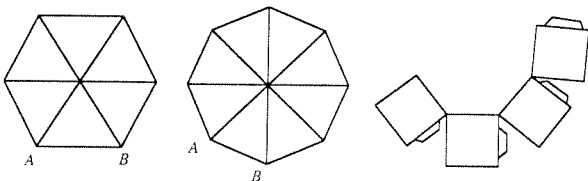
## Kijken naar ruimtelijke vormen

'Kijken naar ruimtelijke vormen' is de titel van het eerste hoofdstuk uit het leerstofpakket *Verkenning in de Ruimte*. In dit hoofdstuk wordt de leerlingen aan de hand van een aantal foto's gevraagd zich een ruimtelijke voorstelling van een situatie te maken. Een voorbeeld is de, misschien reeds bekende, opgave over de N.H. kerk te Smilde.



>a Staan de muren loodrecht op de grond?

Het grondplan van dergelijke kerken is meestal een regelmatige zeshoek of achthoek.



Dit vraagstuk gaat over de zes of acht zijmuren. Modelletjes kunnen daarbij van nut zijn. Je kunt ze ook zelf van papier maken. Voorlopig nemen we geen beslissing over zes- of achthoekig.

- >b Als je je standpunt overal rond de kerk kunt kiezen, dan zijn er verschillende mogelijkheden voor het aantal zichtbare zijmuren. Welke aantallen zijn er mogelijk bij de zeshoek? En welke bij de achthoek?
- >c Teken bij elk van de grondplannen hiervoor het gebied van waaruit alleen muur AB zichtbaar is.
- >d Als je niet buiten dat gebied mag komen, kun je dan beslissen over zeshoek of achthoek?

fig. 12

Foto's zijn echter resultaten van een centrale projectie. Vergeleken met de parallelle projecties die in de andere opgaven gebruikt worden geeft dit bijna altijd een ander plaatje. Evenwijdige lijnen in de voorstel-

ling lopen nu niet altijd evenwijdig in de afbeelding. Deze perspectiefprojectie is ook binnen RuimFig mogelijk. Het ligt dus voor de hand om RuimFig hier ook ondersteuning te laten bieden. In Pakket3D is de centrale projectie niet uitgewerkt.

Als we een achthoekig kerkje willen bekijken dan rijst onmiddellijk de vraag: Hoe voeren we het kerkje als draadmodel in?

Een mogelijkheid is natuurlijk alle hoekpunten in te voeren en vervolgens de goede verbindingen op te geven. Echter, na de bovenstaande voorbeelden doen we dat natuurlijk niet meer. Het kerkje is een regelmatig figuur dat vanuit een bepaald basisfiguur kan ontstaan door draaien rond een as. Wat is de minimale basisfiguur die we nodig hebben? Rond welke as moet er gedraaid worden, met welke hoek, wat is het aantal keer? Vragen die het ruimtelijk voorstellings-vermogen zeker prikkelen.

Een oplossing is:

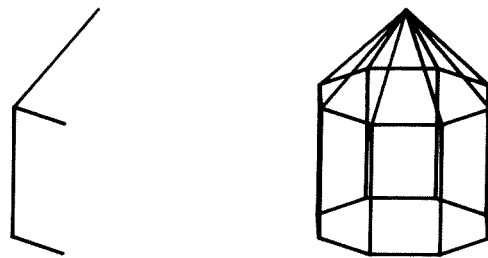


fig. 13a: basisfiguur

fig. 13b: parallelprojectie van de kerk

Voor vragen als in de opgave moeten we de centrale projectie hebben. En vervolgens kijken of het mogelijk is het oog zó te plaatsen, dat we van het kerkje slechts drie zijmuren zien, .. en vier zijmuren?

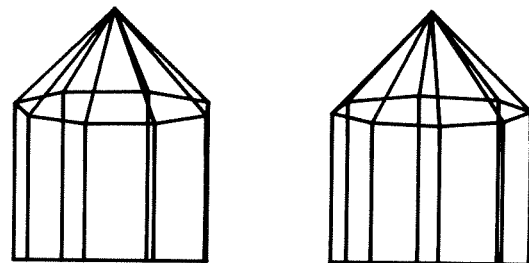


fig. 14a

fig. 14b

Zodoende kwamen we tot een extra formulering van een onderwerp waar de computer ondersteuning kon bieden. Namelijk de overgang van het werken met verschillende concrete objecten naar het interpreteren van foto's en projectiefiguren.

Bovengenoemde opgave over het kunstwerk veranderde dan ook in de volgende opgave:

- 13 > Ga naar PROJECTIES en laat de drie verschillende aanzichten tekenen.
- > Waar moet bij de centrale projectie je oog staan zodat de figuur op het scherm:

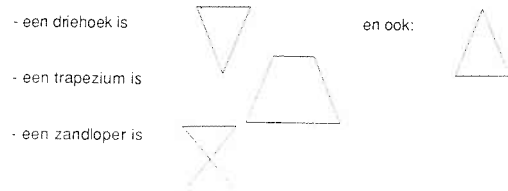


fig. 15

## Conclusie

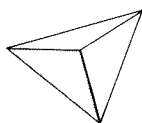
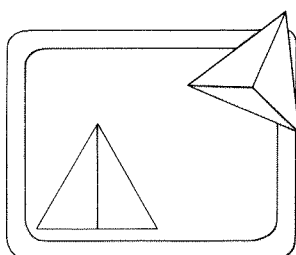
Voorgaande opgaven en voorbeelden geven een idee van hoe de computer ondersteuning kan bieden bij de nieuwe ruimtemeetkunde. Naast de (noodzakelijke) draadmodellen kunnen de programma's Pakket3D en RuimFig als extra hulpmiddelen bijdragen aan de ontwikkeling van het ruimtelijk inzicht.

Binnen het Hawex-project heeft men de voorkeur gegeven aan het menugestuurde RuimFig. De argumenten hiervoor waren de snelheid en het bedieningsgemak. De nadelen van Pakket3D zijn hiermee ook vermeld: het is traag en je moet er tijd in investeren alvorens het resultaat oplevert. De traagheid is vooral een technisch probleem, het aspect van de tijdsinveste-

ring is vooral een principieel punt. Vooralsnog zijn menugestuurde programma's waarbinnen één of twee lessen resultaten mee behaald kunnen worden, het beste in te passen in de praktijk van het onderwijs.

## Literatuur

- [1] Kooij, H. van der: *Dynamiëk in de ruimte*, Nieuwe Wiskrant 7e jaargang nr 4, juli 1988.
- [2] De LCN-LOGO-versie van het programma Pakket3D is gemaakt door het LOGO Centrum Nederland in Nijmegen. Het programma is opgenomen bij het nascholingsmateriaal voor de NIVO-vervolgkursus Wiskunde, dat uitgegeven wordt door het NIB te Zeist.



## Computer-Ondersteuning Wiskunde-Onderwijs

**COWO** is een project dat binnen OW & OC uitgevoerd wordt. Per juni 1989 komen de eerste produkten beschikbaar.

**RUIMFIG** is een ruimtemeetkunde-programma waarmee transformaties op figuren in de ruimte uitgevoerd kunnen worden. Tevens kunnen figuren in verschillende projecties bekeken worden.

**LINPROG** biedt grafische ondersteuning bij 3-dimensionale lineair-programmerings problemen. Tevens kan hiermee de Simplex-methode uitgevoerd worden.

De prijs van één pakket (software + handleiding + lesideeën is  $f$  30,- (exclusief porto- en verzendkosten).

De pakketten kunnen schriftelijk besteld worden bij:

Vakgroep OW & OC  
t.a.v. J. v.d. Voort  
Tiberdreef 4  
3561 GG Utrecht

U dient het (de) bestelnummer(s) van het (de) door u gewenste pakket(ten) te vermelden:

Ruimfig 3.5": 152	Ruimfig 5.25": 151
Linprog 3.5": 153	Linprog 5.25": 154