

Collegediktaat a242

Inleiding Computer Graphics en CAD/CAM

(deel Computer Graphics)

M. Sepers

maart 1988

Technische Universiteit Delft
Faculteit der Technische Wiskunde en Informatica
Vakgroep Technische Informatica
Vakgebied Technische Toepassingen

Inhoudsopgave

0	Inleiding	1
1	Uitvoerapparatuur	
	3	
1.1	Beeldscherm	3
1.1.1	Twee soorten plaatjes	3
1.1.2	De beeldverwerkingseenheid	3
1.1.3	De kathodestraalbuis	3
1.1.4	Het vectorbeeldscherm	5
1.1.5	Het rasterbeeldscherm	7
1.1.6	De direct-view storage tube	7
1.1.7	Het plasma-paneel	9
1.2	Printers	11
1.3	Plotters	13
2	Invoerapparatuur	15
2.1	Inleiding	15
2.2	Het regelpaneel	15
2.3	De lichtpen	15
2.4	De joystick	15
2.5	De muis	15
2.6	Het tablet	17
2.7	De videocamera en de scanner	17
3	Uitvoerprimitieven	19
3.1	Een punt	19
3.2	Een lijn	19
3.3	Een tekst	21
3.4	Een gebied	23
3.4.1	Het scan-line algoritme	23
3.4.2	Het boundary-fill algoritme	27
3.4.3	Het flood-fill algoritme	27
3.5	Prioriteit	27
4	Transformaties	31
4.1	Basis-transformaties	31
4.1.1	Translatie	31
4.1.2	Schaling	31
4.1.3	Rotatie	31
4.2	Samengestelde transformaties	33
4.3	Matrix representaties en homogene coördinaten	35
4.4	Transformaties binnen de framebuffer	37
4.5	Window en viewport	37
4.6	Clipping	39
4.6.1	Clipping van een punt	41
4.6.2	Clipping van een lijn	41
4.6.3	Clipping van een polygoon	47
4.6.4	Clipping van een tekst	47
5	Invoertechnieken	51
5.1	Logische classificatie van invoerapparatuur	51
5.1.1	Het invoeren van de coördinaten van een punt (LOCATOR)	51
5.1.2	Het invoeren van de coördinaten van een serie punten (STROKE)	53
5.1.3	Het invoeren van een tekst (STRING)	53

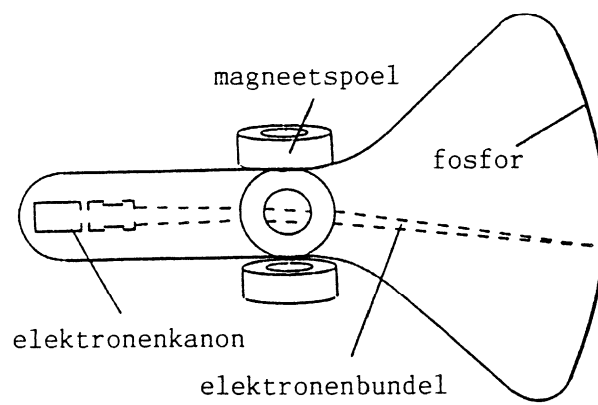
5.1.4	Het invoeren van een getal (VALUATOR)	53
5.1.5	Het aanwijzen van elementen op het scherm (PICK)	55
5.1.6	Het kiezen uit een menu (CHOICE)	57
5.2	Tekentechnieken	59
5.2.1	Plaatsbepalen (positioning)	59
5.2.2	Beperkingen t.a.v. plaats en richting (modular and directional constraints)	59
5.2.3	Attractiesfeer (gravity-field)	61
5.2.4	Rooster, schaal en hulplijn	61
5.2.5	Rubber-band techniek	63
5.2.6	Slepen (dragging)	63
5.3	Invoer toestanden (input modes)	63
6	Segmenten	67
6.1	Inleiding	67
6.2	Displayfile met segmenten	67
6.3	Segment attributen	69
7	Grafische basisprogrammatuur	73
7.1	Inleiding	73
7.2	Basisprogrammatuur (GKS)	73
7.3	Functies binnen GKS	77
7.3.1	Control functies	77
7.3.2	Uitvoer functies	77
7.3.3	Uitvoerattributen	79
	7.3.3.1 Werkstation onafhankelijke attributen	79
7.3.4	Tranformatie-funcities	81
	7.3.4.1 Tranformatie naar genormaliseerde toestelcoördinaten	81
	7.3.4.2 Transformatie naar apparaatcoördinaten	81
7.3.5	Segment functies	81
	7.3.5.1 Functies die het werken met segmenten mogelijk maken	81
	7.3.5.2 Segment attributen	83
7.3.6	Invoer functies	83
	7.3.6.1 Initialiseren van de invoerapparaten	85
	7.3.6.2 Het instellen van de manier waarop de gegevens worden ingevoerd	85
	7.3.6.3 Invoer in request mode	85
	7.3.6.4 Invoer in sample mode	85
	7.3.6.5 Invoer in event mode	85
7.3.7	Metafile functies	87
7.3.8	Inquiry functies	87
7.3.9	Utility functies	87
7.3.10	Foutafhandeling	89
7.4	Programma met GKS-functies	89

0 Inleiding.

Grafische afbeeldingen kunnen een efficiënt middel voor het overdragen van informatie zijn. Op veel plaatsen wordt dan ook nuttig gebruik gemaakt van computer graphics, b.v.:

- in de industrie bij het interactief ontwerpen en tekenen
- in de zakelijke sector bij presentaties m.b.v. grafieken voor omzet, kosten, enz.
- in de computer kunst bij het maken van abstracte ontwerpen m.b.v. wiskundige functies of het maken van plaatjes op het scherm m.b.v. schildertechnieken
- in de vliegtuigindustrie bij de training van piloten (flight simulator)
- in de filmindustrie bij het maken van tekenfilms (animatie)
- in de medische sector bij het weergeven van dwarsdoorsneden (scans) van het menselijk lichaam
- in de astronomie bij het onderzoek van melkwegstelsels
- in de cartografie bij het maken van kaarten
- in de huiselijke kring bij computerspelletjes

In het college a242 (deel computer graphics) zullen enkele grafische basistechnieken voor dit soort toepassingen worden behandeld vanuit het gezichtspunt van de informatica.



Figuur 1.1 Het principe-schema van de CRT.

1. Uitvoerapparatuur.

1.1 Beeldscherm.

1.1.1 Twee soorten plaatjes.

De plaatjes, die m.b.v. een computer gemaakt worden, kunnen in twee groepen worden ingedeeld:

1. Plaatjes, die alleen uit lijnen bestaan (line drawings).
2. Plaatjes, die zijn opgebouwd uit vlakken in verschillende kleurschakeringen (continuous-tone images).

Deze plaatjes zien er niet alleen heel verschillend uit, maar vragen ook een heel verschillende techniek om ze te maken. De plaatjes uit de eerste groep zijn in verschillende opzichten makkelijker te verkrijgen, omdat het algoritme om ze op te bouwen eenvoudiger is, de hoeveelheid gegevens om ze te beschrijven minder is en de apparatuur om ze af te beelden eenvoudiger kan zijn. Voor de plaatjes uit de tweede groep is een beeldscherm nodig, dat meerdere kleurschakeringen (eventueel grijstinten) kan weergeven.

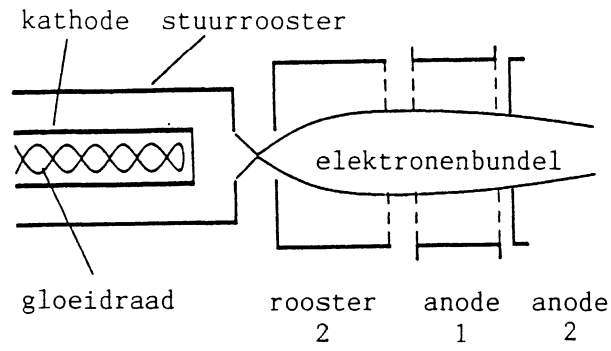
1.1.2 De beeldverwerkingseenheid.

De beeldverwerkingseenheid (display processing unit - DPU) zorgt voor de vertaling van de beeldgegevens in de computer naar de elektrische signalen die nodig zijn om het plaatje weer te geven op het beeldscherm. De DPU verzorgt ook de afbeelding van letters en andere symbolen. Naast deze primaire taken zijn sommige beeldverwerkingseenheden voor grafische toepassingen uitgerust met extra hardware om bijvoorbeeld transformaties uit te kunnen voeren. Met deze hardware-matige aanpak kan t.o.v. een software-matige benadering een aanmerkelijke snelheidswinst worden behaald.

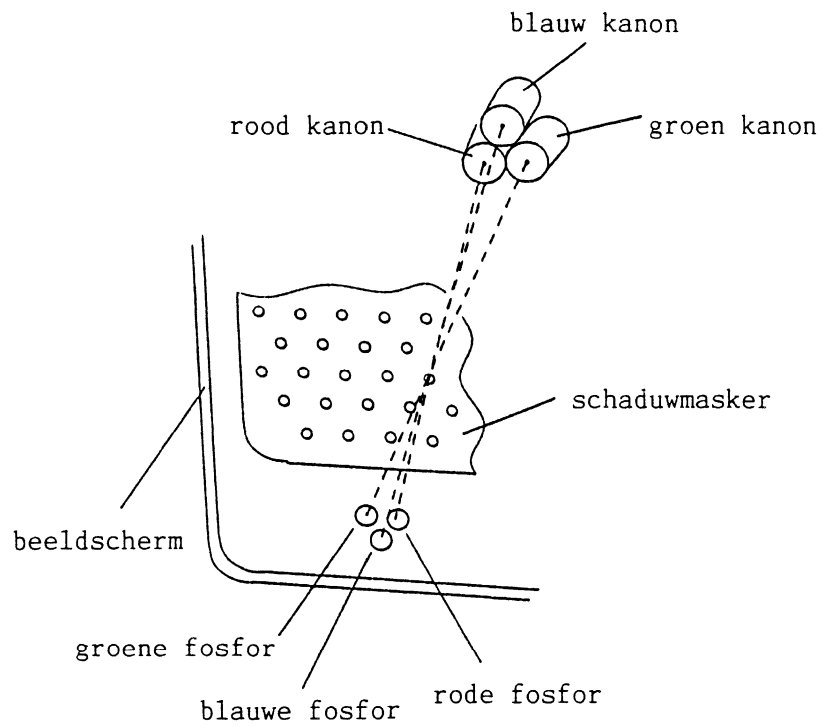
1.1.3 De kathodestraalbuis.

De Whirlwind I computer van het Massachusetts Institute of Technology (MIT) was in 1950 het eerste systeem met een computer gestuurd beeldscherm. Men gebruikte daarvoor een elektronen- of kathodestraalbuis (cathode ray tube - CRT), omdat deze elektrische signalen snel kan omzetten in zichtbare beelden. In die tijd gaf de CRT nog geen scherp beeld. Bovendien was het beeld klein en het apparaat duur. In de loop der jaren werd de CRT verder ontwikkeld en verbeterd. De speciale eigenschappen van de CRT hebben de ontwikkeling van de computer graphics-techniek in belangrijke mate bepaald.

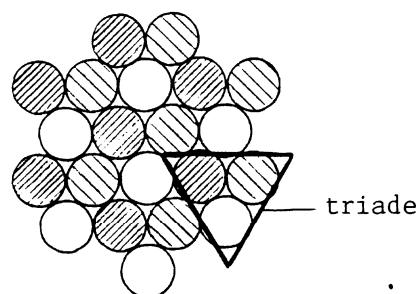
De CRT (figuur 1.1) bestaat in zijn meest elementaire uitvoering uit een afgesloten glazen buis (vacuüm) met een fosforscherm, een elektronenkanon en een afbuigingssysteem. Het elektronenkanon bevindt zich binnen de glazen buis. Bij het kanon is de buis smal, meer van het kanon af wordt de buis breder.



Figuur 1.2 Het elektronenkanon van de CRT.



Figuur 1.3 De CRT met schaduwmasker.



Figuur 1.4 Een pixel omvat meerdere triaden.

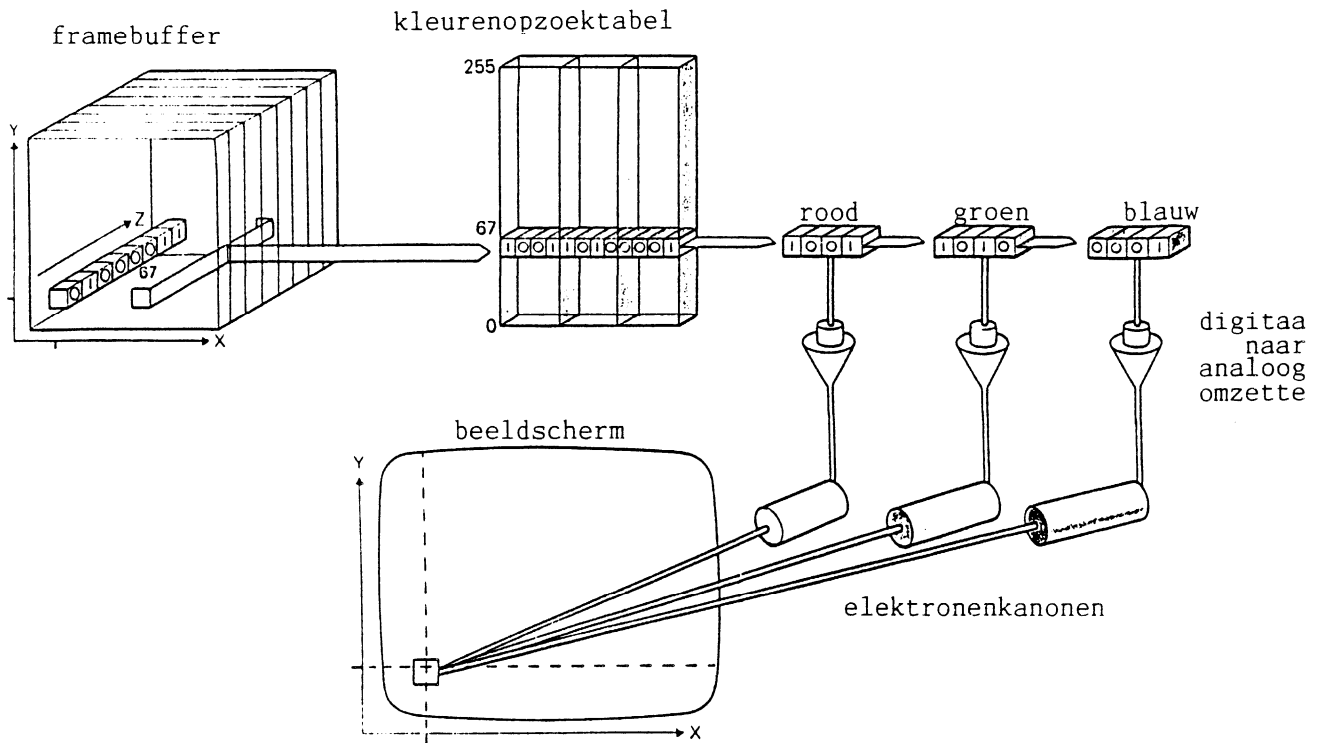
Het elektronenkanon (figuur 1.2) bestaat uit een kathode, die de elektronen emitteert en een anode, die de elektronen aantrekt. Door een groot spanningsverschil van enkele duizenden volts tussen kathode en anode krijgen de elektronen een grote snelheid. De anode maakt meestal deel uit van een focuseringssysteem. Het elektrostatisch veld binnen dit systeem is zodanig, dat er een soort lenswerking ontstaat. De elektronen worden binnen een smalle bundel (straal) geconcentreerd. De bundel is op zijn smalst, als de elektronen het scherm treffen. Tussen de kathode en het focuseringssysteem is een rooster geplaatst. Door de spanning van dit rooster t.o.v. de kathode te veranderen, kan de intensiteit (het aantal elektronen per tijdseenheid) van de elektronenbundel gevarieerd worden.

Het afbuigingssysteem bevindt zich buiten de glazen buis. Het kan bestaan uit twee paar magneetspoelen. Door de stroom door de spoelen te veranderen kan de afbuiging zowel in horizontale als in verticale richting gewijzigd worden.

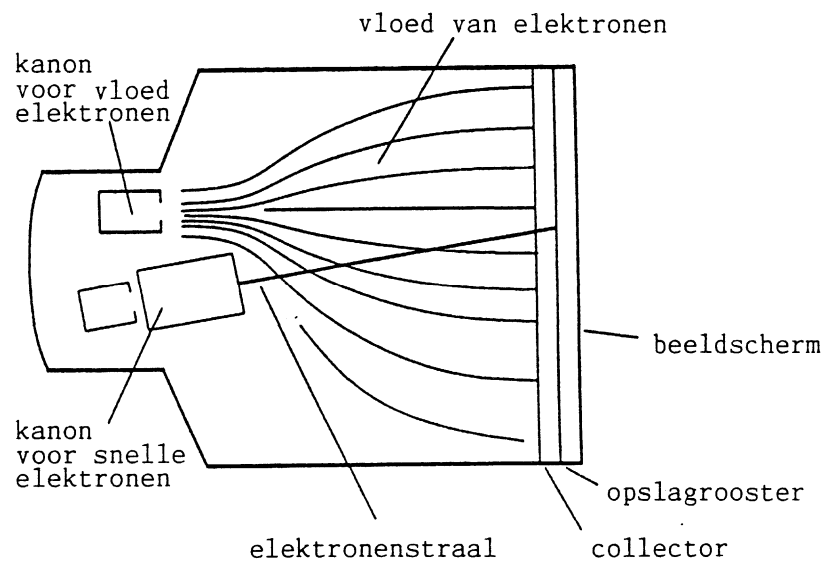
De fosforlaag, die aan de binnenkant van de glazen buis wordt aangebracht, gaat gloeien (fosforescentie) als de elektronenbundel er op valt. Als de plaats van de elektronenbundel verandert, gloeit de fosfor nog even na. Door stoffen aan de fosfor toe te voegen kan de nagloeitijd gewijzigd worden. M.b.v. toevoegingen kunnen ook verschillende kleuren worden geëmitteerd. Voor het maken van een plaatje moet de bundel zich over het scherm verplaatsen. Vanwege de beperkte nagloeitijd moet het plaatje steeds opnieuw getekend worden. Een te lange nagloeitijd geeft hinderlijke na-gloeieffecten (smearing) bij bewegende plaatjes. Een te korte nagloeitijd geeft een flikkerend beeld. In de praktijk streeft men naar een opfrisfrequentie (refresh rate) van minstens 30 maal per seconde om een voldoende snelle en stabiele afbeelding te kunnen maken.

1.1.4 Het vectorbeeldscherm.

Op een vectorbeeldscherm (line drawing of vector display) wordt de elektronenstraal langs lijnstukken gestuurd. Een driehoek bijvoorbeeld kan getekend worden door de coördinaten van de hoekpunten in te voeren in de computer. Om de driehoek op het scherm te krijgen moeten deze coördinaten in een displayfile (een lijst met instructies voor de DPU) worden geplaatst. De displayfile wordt opgeslagen in een speciaal regeneratiegeheugen (refresh buffer). Om de lijnstukken op het scherm te tekenen moeten de x-afbuiging, de y-afbuiging en de intensiteit worden ingesteld. De DPU moet de digitale waarden uit de refresh buffer dus omzetten in analoge spanningen voor de CRT. Dit wordt gedaan m.b.v. een vectorgenerator. Gebogen lijnen kunnen worden getekend door deze met kleine lijnstukken te benaderen. Na elke regeneratiecyclus (refresh cycle) moet de displayfile om bovengenoemde reden weer van voren af aan worden afgewerkt. Een displayfile met tienduizenden lijnstukken is niet ongebruikelijk.



Figuur 1.5 De kleurschakering van een pixel op een rasterbeeldscherm.



Figuur 1.6 De direct-view storage tube.

1.1.5 Het rasterbeeldscherm.

Op een rasterbeeldscherm (rasterdisplay) wordt de elektronenstraal net als bij een gewone televisie langs horizontale lijnen (scan-lines) gestuurd. Bovenaan het scherm beginnend wordt naar de onderkant toe gewerkt. Om in kleur te kunnen tekenen worden op het scherm kleine puntjes fosfor, die rood, groen en blauw licht emitteren, aangebracht. De puntjes worden in gelijkzijdige driehoekjes t.o.v. elkaar geplaatst (triaden). De kleuren rood, groen en blauw zijn de drie additieve primaire kleuren. Andere kleuren kunnen worden verkregen door deze drie kleuren in een bepaalde verhouding te mengen. Additieve menging komt tot stand wanneer de kleuren afkomstig zijn van een lichtbron. Voor elke kleur fosfor wordt een apart elektronenkanon gebruikt. Vlak achter het beeldscherm wordt een schaduwmasker (shadow-mask) geplaatst. Dit is een metalen scherm met openingen, waardoor een bepaalde kleur fosfor alleen door het bijbehorende kanon bestraald kan worden (figuur 1.3). De triaden worden samengevoegd tot pixels (picture elements) (figuur 1.4), die apart aangestuurd kunnen worden.

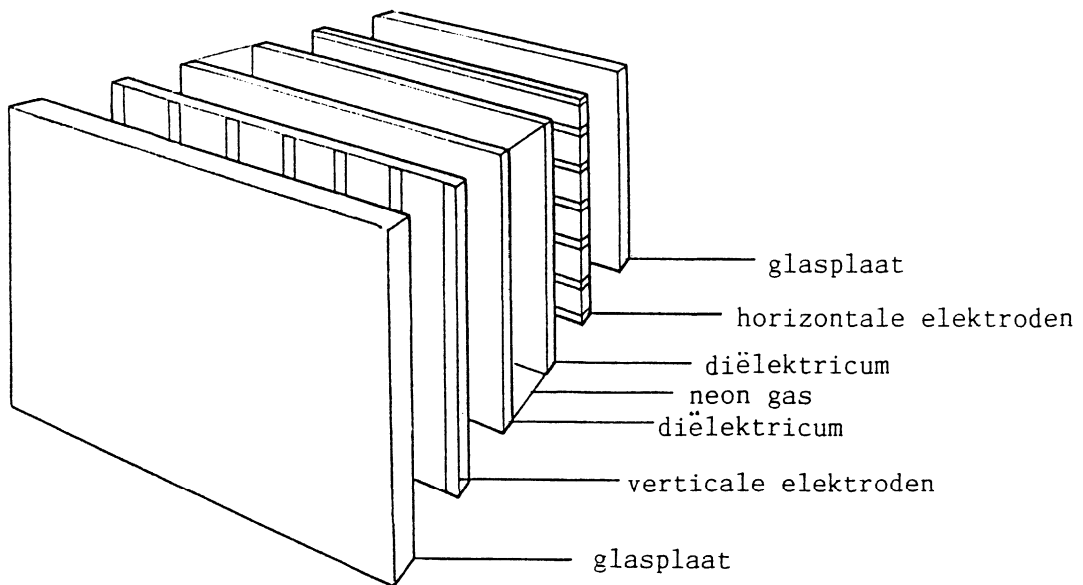
Kleurenplaatjes kunnen gemaakt worden door de intensiteit van de drie kanonnen te moduleren. Daartoe wordt het plaatje opgeslagen in een speciaal beeldgeheugen, n.l. de frame buffer (figuur 1.5). De frame buffer is een groot RAM (Random-Access Memory), waarin voor de kleur van elk pixel een aantal bits zijn gereserveerd. Met 8 bits per pixel kunnen 256 ($= 2^8$) kleuren worden vastgelegd. Vaak maakt men gebruik van een kleurenopzoektabel (look-up table (LUT)) (figuur 1.5). Als een kleurschakering in de kleurenopzoektabel met 12 ($= 3 \times 4$) bits is vastgelegd, kan er gekozen worden uit een palet van 4096 ($= 2^{12}$) kleuren.

Een plaatje op een rasterdisplay kan gezien worden als een matrix van pixels, die allemaal op de juiste kleurschakering kunnen worden ingesteld. Het aantal adresseerbare pixels in horizontale en verticale richting, wordt het oplossend vermogen of de resolutie van het rasterbeeldscherm genoemd. De resolutie is strikt genomen het aantal pixels per eenheid van schermlengte.

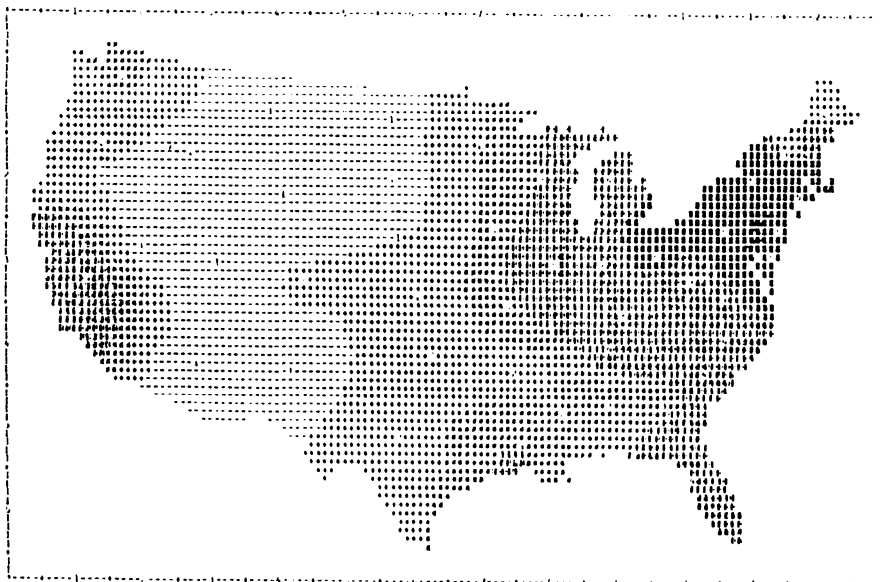
Voor het manipuleren van honderdduizenden pixels zijn geheugens en verwerkingseenheden met een grotere capaciteit nodig dan voor het werken met lijnstukken. Maar terwijl vectorsystemen vele jaren monochroom bleven, hield de rastertechnologie zich bijna vanaf het begin bezig met kleur. Een rasterbeeldscherm biedt andere toepassingsmogelijkheden, zoals het afbeelden van gekleurde oppervlakken van voorwerpen.

1.1.6 De direct-view storage tube.

Bij de direct-beeld opslagbuis (direct-view storage tube - DVST) is een fijnmazig opslagrooster (storage grid) vlak achter het scherm aangebracht (figuur 1.6). Het opslagrooster



Figuur 1.7 Het plasma-paneel.



Figuur 1.8 Plaatje, dat m.b.v. een line-printer is gemaakt.
De gradaties worden verkregen door een geschikte
keuze van de tekens. Op de donkerste plaatsen zijn
meerdere tekens over elkaar getypt.

is bedekt met een dielektricum en heeft aanvankelijk een uniforme negatieve lading. Achter het opslagrooster bevindt zich nog een tweede rooster. Dit rooster, dat de collector wordt genoemd, heeft een positieve spanning t.o.v. het opslagrooster. Net als op een vectorbeeldscherm wordt de elektronenstraal langs lijnstukken gestuurd, maar nu treft de straal het opslagrooster. Door de hoge snelheid van de elektronen worden de elektronen van het opslagrooster losgeschoten. Deze elektronen gaan vervolgens naar de collector. Het resultaat is, dat de elektronenstraal een patroon van minder negatieve lading op het opslagscherm schrijft. Een tweede elektronenkanon straalt een vloed van langzame elektronen uit. Deze elektronen, die gelijkmatig op het scherm zijn gericht, worden afgebogen door het ladingspatroon op het opslagrooster en dat brengt het beeld over op het met fosfor bedekte scherm.

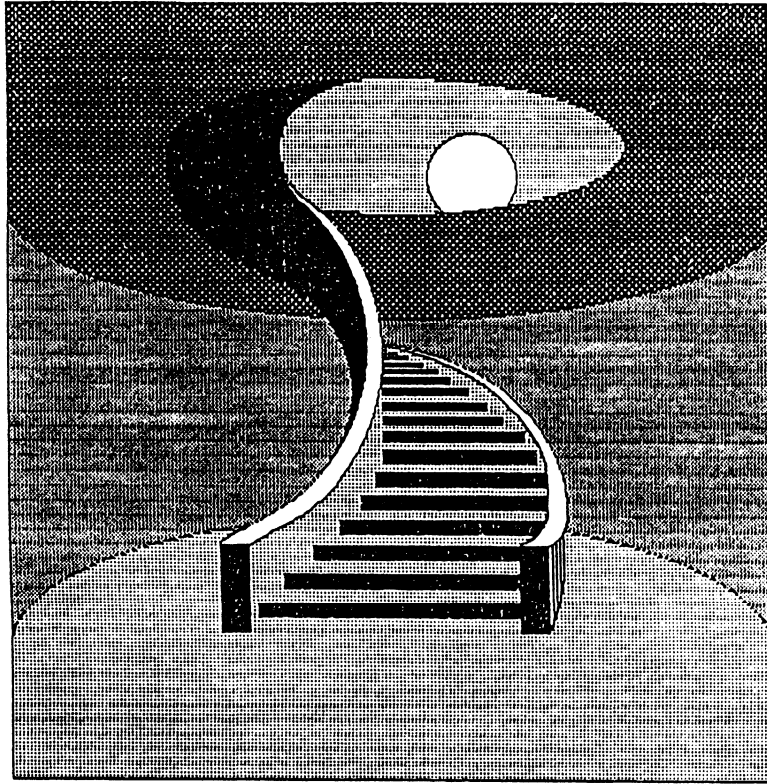
Dat de complexere buis meer kost, wordt gecompenseerd door de besparingen op de bijkomende elektronica. Een plaatje dat op het scherm verschijnt, wordt vastgehouden door het scherm zelf. Een refresh-buffer is daarom niet nodig en ook de schrijfsnelheid kan lager zijn. Als het plaatje op het scherm moet worden veranderd wordt het hele beeld gewist door een positieve spanning op het opslagrooster aan te brengen, waardoor de elektronenvloed het ladingspatroon verwijdert.

De DVST heeft een stabiel beeld met een hoog oplossend vermogen en kan een groot aantal beeldgegevens vasthouden. Een groot nadeel van de DVST is dat, als een enkel lijnstuk verwijderd moet worden, het scherm moet worden schoongemaakt en het hele plaatje opnieuw moet worden opgebouwd. Een DVST is daarom niet geschikt voor interactieve toepassingen en bewegende beelden.

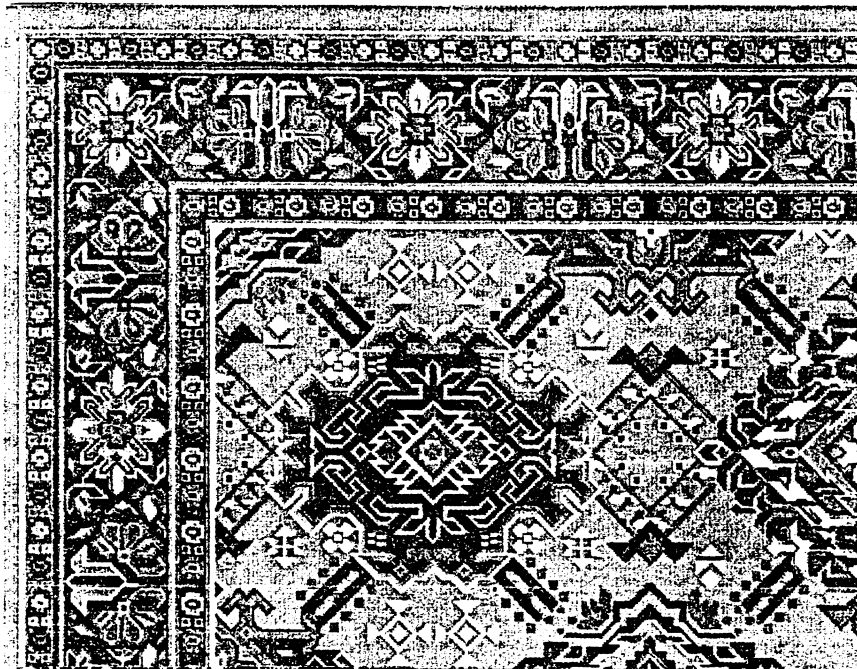
1.1.7 Het plasma-paneel.

Het plasmapaneel (plasma-paneel) bestaat uit twee evenwijdige glasplaten (figuur 1.7), die op korte afstand van elkaar zijn geplaatst. Op de glasplaten zijn aan de binnenzijde, op korte afstand van elkaar, evenwijdige elektroden aangebracht. Op de ene glasplaat lopen de elektroden horizontaal en op de andere verticaal. De elektroden zijn afgedekt met een dielektrisch materiaal. Binnen de afgesloten ruimte tussen de glasplaten bevindt zich neon gas. Tussen de horizontale en verticale elektroden wordt een wisselspanning met een hoge frequentie aangebracht. Met behulp van pulsjes op deze wisselspanning kan plaatselijk een gloeiontlading gestart en gestopt worden. Het plasmapaneel kan daarom gezien worden als een matrix van kleine neonlampjes, die allemaal apart aan of uit kunnen worden gedaan. Schakelsnelheden van 10 microseconden per lampje zijn mogelijk. Eventueel kunnen meerdere lampjes tegelijkertijd aangestuurd worden.

Het plasmapaneel heeft een volkomen stabiel beeld. Net als bij de DVST wordt een plaatje, dat op het scherm verschijnt vastgehouden door het scherm zelf. Maar bij een plasmapaneel kunnen veranderingen worden aangebracht zonder dat het



Figuur 1.9 Afdruk m.b.v. een matix-printer.



Figuur 1.10 Tapijtmotief dat m.b.v. een ink-jet-printer is afgedrukt.

plaatje opnieuw moet worden opgebouwd. De resolutie is echter kleiner dan die van de DVST, b.v. 25 lampjes per cm. De oppervlakte van het paneel kan wel enkele vierkante meters groot zijn bij een relatief geringe dikte.

1.2 Printers.

Alhoewel printers oorspronkelijk ontworpen zijn om tekst op papier af te drukken, zijn ze tegenwoordig ook goed te gebruiken voor grafische uitvoer. Een printer drukt een plaatje af op een manier, die doet denken aan een rasterbeeldscherm. Van links naar rechts en van boven naar beneden. Daarom zal een plaatje meestal tijdens het opbouwen opgeslagen worden in een buffer. Als het plaatje klaar is, gaat het naar de printer. Figuur 1.8 laat een plaatje zien, dat m.b.v. een line-printer is gemaakt. Sommige printers hebben speciale voorzieningen voor grafische toepassingen, waardoor de snelheid en de kwaliteit aanzienlijk beter worden.

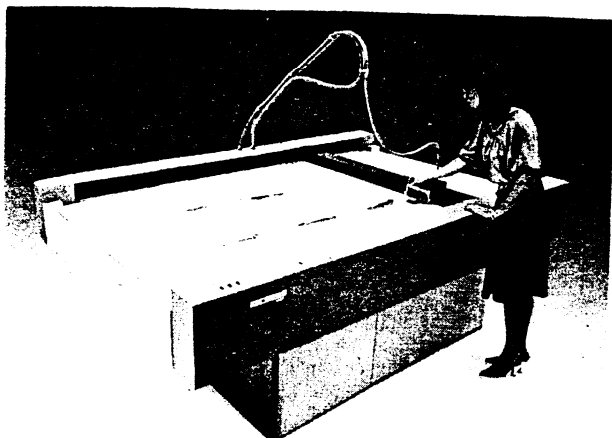
Er is een heel scala aan technieken ontwikkeld om de inkt op het papier te krijgen. Hieronder worden enkele printers genoemd, die geschikt zijn voor grafische toepassingen.

Een matrixprinter heeft een afdrukkop met een rij dunne pennen, die afzonderlijk gestuurd kunnen worden. Door een elektromagnetische kracht kunnen de pennen een inktlint tegen het papier drukken. Als de afdrukkop langs het papier schuift, ontstaat een matrix van kleine ronde puntjes inkt op korte afstand van elkaar (figuur 1.9). Door heel kleine verschuivingen in horizontale en verticale richting kunnen letters van goede kwaliteit worden afgedrukt. Met behulp van gekleurd inktlint kunnen ook kleurschakeringen worden verkregen. Men gebruikt hiervoor inkt in de drie subtractieve primaire kleuren (cyaan, magenta en geel) en zwart voor de nuanciering. In dit geval de subtractieve kleuren, omdat inkt op papier bepaalde kleuren van het erop vallende licht absorbeert.

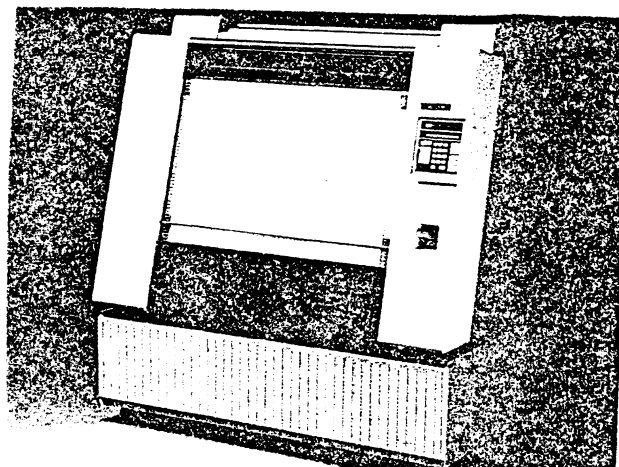
Bij een ink-jet printer worden fijne druppels inkt tegen het papier gespoten. De elektrisch geladen inkt wordt door een electrostatisch veld naar de juiste plaats gestuurd. Het afdrukken in kleur gaat eenvoudig door met meerdere spuitmondjes tegelijkertijd in de drie subtractieve kleuren te spuiten (figuur 1.10).

In een laserprinter wordt een laserstraal m.b.v. een spiegeltje gestuurd. De laserstraal brengt een patroon van negatieve lading aan op een roterende trommel, die bedekt is met een laagje foto-elektrostatisch materiaal (selenium). Daarna wordt m.b.v. een positief geladen inkt (toner) een afdruk gemaakt op papier. D.m.v. verhitting wordt de inkt vervolgens gefixeerd.

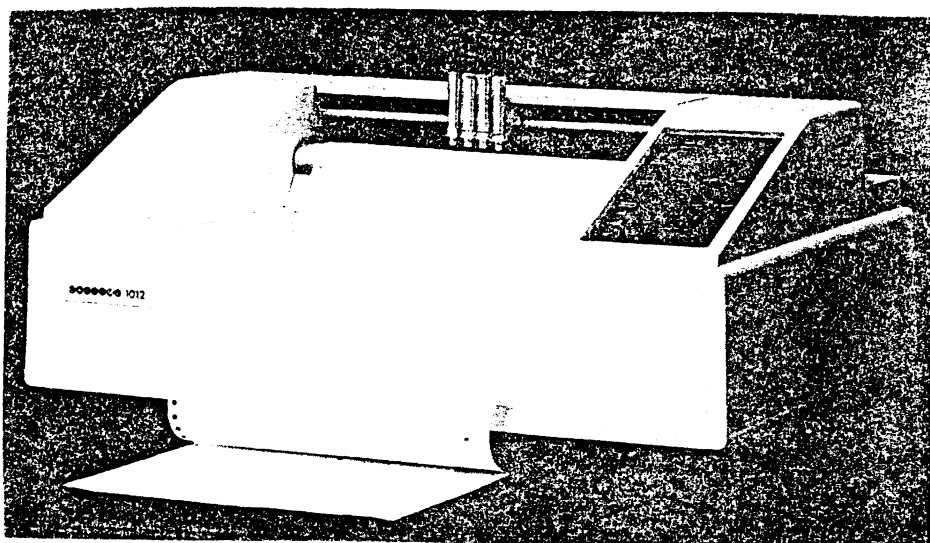
De thermotransferprinter heeft een afdrukkop met een rij puntjes, die afzonderlijk warm gemaakt kunnen worden. Het lint heeft een soort inkt op wasbasis, die los komt op



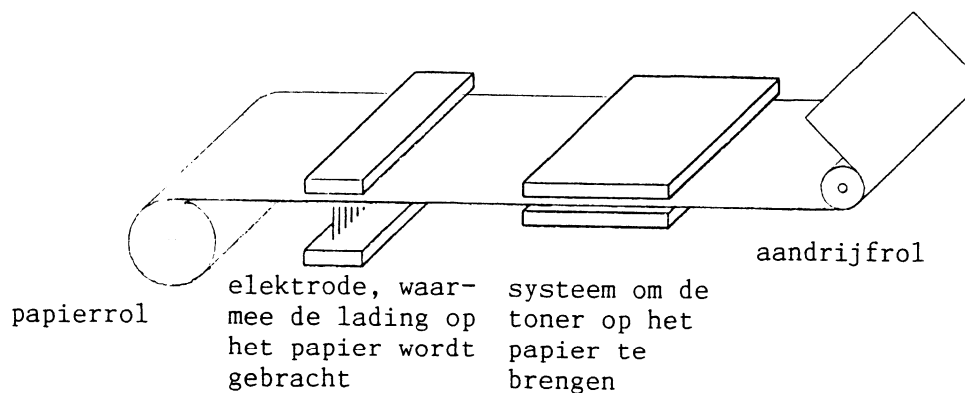
Figuur 1.11 Flatbed-plotter



Figuur 1.13 Beltbed-plotter



Figuur 1.12 Drum-plotter



Figuur 1.14 Het principe-schema van de elektrostatische plotter.

plaatsen, waar het warm wordt. De inkt blijft dan kleven aan het papier.

Bijna alle printers werken met een matrix van puntjes. De patronen voor verschillende lettertypen zijn vaak in het apparaat zelf opgeslagen. De resolutie, d.i. het aantal puntjes per lengte-eenheid, is de laatste jaren sterk toegenomen tot wel 160 puntjes per cm. Tegenwoordig kan m.b.v. een printer een afdruk van goede kwaliteit worden gemaakt.

1.3 Plotters.

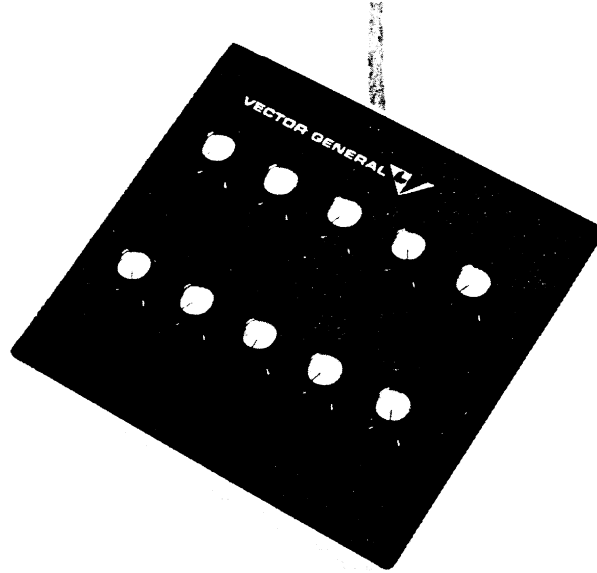
De manier waarop pen-plotters een plaatje opbouwen, doet denken aan een vectorbeeldscherm. De pen verbindt punten in willekeurige richtingen. D.m.v. servomotoren wordt de pen naar een andere positie gestuurd. M.b.v. een elektromagneet wordt de pen opgetild en neergezet. Sommige plotters staan alleen bewegingen in kleine stapjes toe, terwijl andere (meestal snellere) in één beweging de pen naar de nieuwe positie brengen. Er kunnen pennen in verschillende soorten (tekenpen, balpen, viltstift), dikten en kleuren worden gebruikt. Vaak kan vanuit een programma een commando gegeven worden om een pen te verwisselen.

Bij een flatbed-plotter (figuur 1.11) ligt het papier vlak uitgespreid op een tafel. Het papier wordt meestal op zijn plaats gehouden door een zuigsysteem of een elektrostatisch veld in de tafel. De pen kan gelijktijdig in twee richtingen verplaatst worden. Een dwarsbalk kan van de ene naar de andere kant van de plotter bewegen, terwijl de pen heen en weer langs de balk gaat.

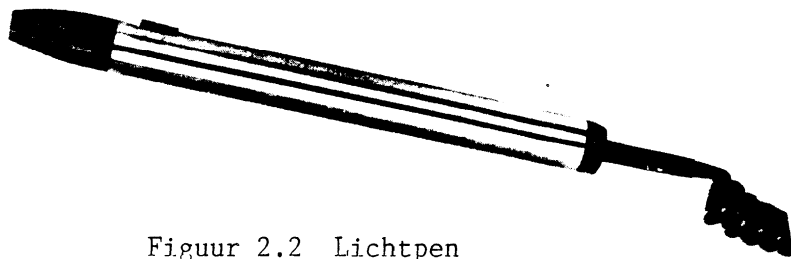
In een drum-plotter (figuur 1.12) is het papier vastgeklemd op een draaibare rol en in een beltbed-plotter (figuur 1.13) zit het op een brede lopende band. Bij deze plotters kan de pen slechts in één richting langs een balk heen en weer bewegen. Om in een andere richting te tekenen moet tegelijkertijd de rol of de band heen en weer bewogen worden.

Bij een elektrostatische plotter (figuur 1.14), loopt het papier onder een soort kam met fijne metalen tanden. Via de tanden wordt er een patroon van negatieve lading op het papier gebracht. Het papier passeert vervolgens een toner, waardoor het patroon zichtbaar wordt. Door dit proces te herhalen met toner in de drie subtractieve primaire kleuren kan er ook een kleurenafdruk worden gemaakt.

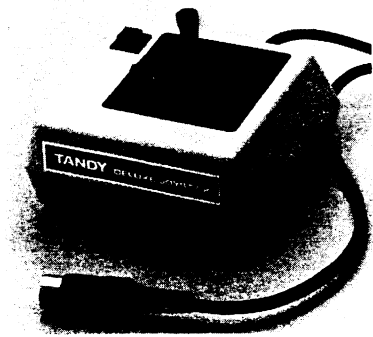
Plotters zijn vaak voorzien van hardware om snel verschillende lijnsoorten (streepjes, puntjes) en basisfiguren (cirkel, ellips) te kunnen tekenen. Plotters hebben in het algemeen een hogere resolutie dan printers.



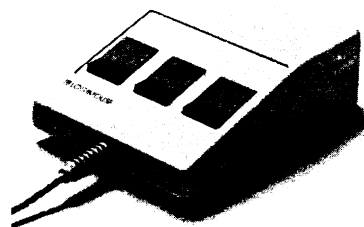
Figuur 2.1 Regelpaneel



Figuur 2.2 Lichtpen



Figuur 2.3 Joystick



Figuur 2.4 Muis

2. Invoerapparatuur.

2.1 Inleiding.

Naast het toetsenbord, dat geschikt is om tekst en getallen in te voeren, bestaat er ook invoerapparatuur om iets aan te wijzen (pointing) of een plaats te bepalen (positioning). Hoewel veel toetsenborden speciale toetsen hebben voor het sturen van een cursor (symbool op het scherm voor het aangeven van de positie), is er voor grafische toepassingen behoefte aan apparaten die dat meer rechtstreeks en handzamer doen. D.m.v. een directe visuele terugkoppeling kan ook als een invoerapparaat niet lineair is, d.w.z. de verandering van de invoercoördinaat is niet evenredig met de verplaatsing van de hand, een goede bestuurbaarheid worden verkregen.

2.2 Het regelpaneel.

Op een regelpaneel (figuur 2.1) zitten regelknoppen (control dials) waarmee speciale (samengestelde) opdrachten gegeven kunnen worden. Door aan een knop te draaien kan bijvoorbeeld een cursor op het scherm verplaatst worden of een afgebeeld voorwerp geroteerd worden.

2.3 De lichtpen.

M.b.v. een lichtpen (figuur 2.2) kan op het scherm een lichtgevend element, bijvoorbeeld een punt, een lijn of een symbool, aangewezen worden. Als tijdens de refresh cycle de elektronenstraal de plaats van de pen bereikt en de knop op de pen ingedrukt is, krijgt de processor een signaal en moet het aangewezen element worden bepaald. De lichtpen wordt weinig meer gebruikt, omdat er tegenwoordig goede andere mogelijkheden bestaan, die goedkoper en bovendien minder vermoeiend voor de arm zijn.

2.4 De joystick.

De joystick (figuur 2.3) is een stuurknuppel, waarmee een cursor op het scherm verplaatst kan worden. De richting, waarin de knuppel wordt gedrukt bepaalt de richting van de verplaatsing. Vaak gaat de verplaatsing sneller als de knuppel verder wordt gedrukt. Hierdoor kan een snelle positionering worden verkregen.

2.5 De muis.

M.b.v. een muis (figuur 2.4) kan op het scherm een cursor of kruisdraad gestuurd worden. De muis is een doosje met aan de onderkant een voorziening, die de verplaatsing in twee loodrechte richtingen registreert. De grootte van de verplaatsing in beide richtingen wordt d.m.v. een evenredig aantal pulsjes doorgegeven aan de computer. Bij een volgende refresh cycle verschijnt de cursor dan op de nieuwe positie. Boven op de muis zitten een of meer druktoetsen, waarmee opdrachten gegeven kunnen worden. Als de muis wordt opgetild of slipt, wordt er geen verplaatsing geregistreerd. Door



Figuur 2.5 Digitizer

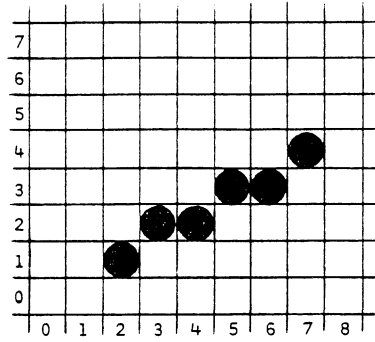
deze eigenschap is de muis niet geschikt voor bijvoorbeeld het bepalen van de coördinaten van punten op een tekening.

2.6 Het tablet.

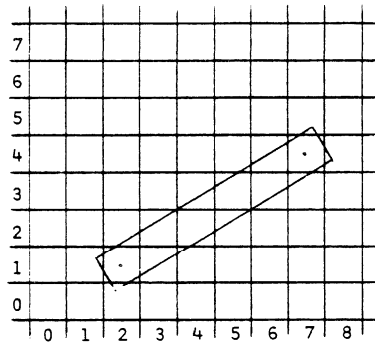
Het tablet of de digitizer (figuur 2.5) is een vlakke plaat, waarop d.m.v. een handcursor of een stylus, d.i. een soort tekenpen, een plaats aangewezen kan worden. Een tablet heeft een absolute besturing, omdat dezelfde plaats steeds met dezelfde coördinaten correspondeert. Door deze eigenschap is een tablet geschikt voor het bepalen van de coördinaten van punten op een tekening, die op het tablet is vastgemaakt (digitaliseren). Voor het bepalen van de coördinaten worden verschillende methoden toegepast: elektrisch, elektromagnetisch, akoestisch. De handcursor en de stylus hebben één of meerdere druktoetsen, waarmee opdrachten gegeven kunnen worden.

2.7 De videocamera en de scanner.

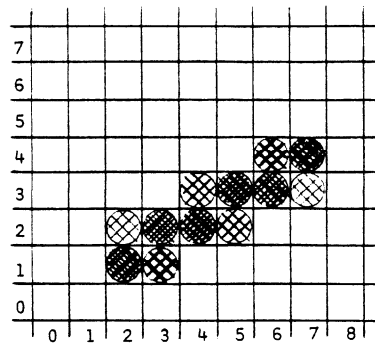
M.b.v. een videocamera en een scanner kan direct een rasterbeeld worden ingevoerd. Er is dan nog geen geometrische beschrijving van het beeld. Wel is er software ontwikkeld om een dergelijk beeld te herkennen en te manipuleren. Het vakgebied hiervoor is beeldverwerking.



Figuur 3.1 Deel van een rasterbeeldscherm met een lijnstuk van pixel P(2,1) naar pixel Q(7,4).



Figuur 3.2 Deel van een rasterbeeldscherm met een lijnstuk (strook) van pixel P(2,1) naar pixel Q(7,4).



Figuur 3.3 Deel van een rasterbeeldscherm met een lijnstuk van pixel P(2,1) naar pixel Q(7,4), waarbij gebruik is gemaakt van een anti-aliasing methode.

3. Uitvoerprimitieven.

3.1 Een punt.

Een punt wordt op een CRT afgebeeld door de intensiteit van de elektronenstraal op een bepaalde plaats te verhogen, waardoor de fosfor op het scherm in een klein gebied gaat gloeien. Met een vectorbeeldscherm kan dit tot stand worden gebracht door een opdracht om een punt af te beelden in de displayfile op te nemen. Bij een rasterbeeldscherm moet aan een pixel in de framebuffer de juiste waarde worden gegeven.

Stel dat met de procedure `set_pixel (x,y,c)` een pixel op plaats (x,y) de kleur c (c is geheel) krijgt. Omdat een pixel het kleinste aanstuurbare element is, kunnen voor x en y gehele getallen worden genomen.

3.2 Een lijn.

Met een vectorbeeldscherm kan een lijn getekend worden door een opdracht om een lijnstuk af te beelden in de displayfile op te nemen. De elektronenstraal gaat dan rechtstreeks van het beginpunt naar het eindpunt van het lijnstuk. Bij een rasterbeeldscherm gaat de elektronenstraal langs de scan-lines en moet een lijn door een aantal pixels worden gerepresenteerd (scan conversion).

Stel dat een lijnstuk getekend moet worden van pixel $P(x_1, y_1)$ naar pixel $Q(x_2, y_2)$. Bij het z.g. simple DDA (Digital Differential Analyzer) algoritme onderscheidt men voor het bepalen van de tussenliggende pixels twee gevallen:

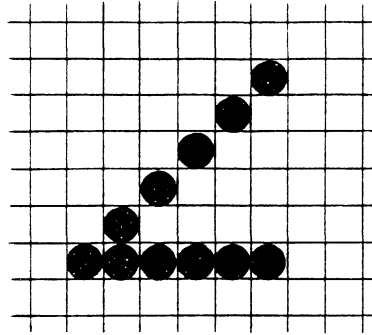
1. Als $|x_2 - x_1| > |y_2 - y_1|$ dan in $|x_2 - x_1|$ stappen van P naar Q (de stapgrootte in de x -richting is 1).
2. Als $|x_2 - x_1| \leq |y_2 - y_1|$ dan in $|y_2 - y_1|$ stappen van P naar Q (de stapgrootte in de y -richting is 1).

Tussenliggende pixels kunnen m.b.v. de volgende formules berekend worden:

$$x_{i+1} = x_1 + (x_2 - x_1)/n$$

$$y_{i+1} = y_1 + (y_2 - y_1)/n$$

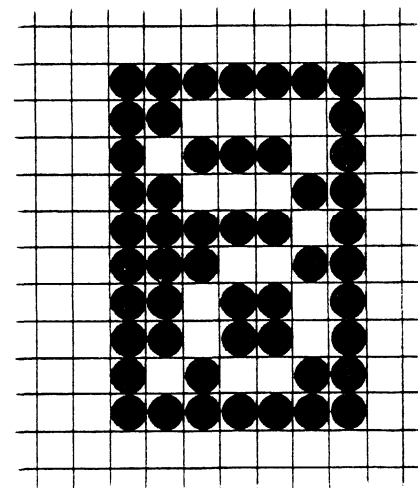
Hierin is n het aantal stappen.



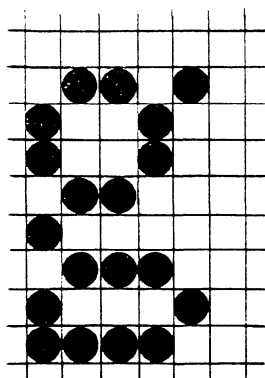
Figuur 3.4 Deel van een rasterbeeldscherm met een horizontaal en een schuin lijnstuk. Bij het schuine lijnstuk is het aantal pixels per lengte-eenheid kleiner, waardoor de helderheid minder is.

5	0	0	0	0	0	0
4	0	1	1	0	1	0
3	1	0	0	1	0	0
2	1	0	0	1	0	0
1	0	1	1	0	0	0
0	1	0	0	0	0	0
-1	0	1	1	1	0	0
-2	1	0	0	0	1	0
-3	1	1	1	1	0	0
	0	1	2	3	4	5

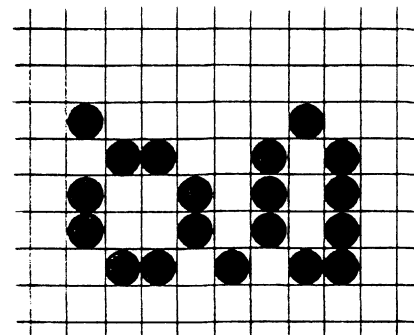
Figuur 3.5 Bitpatroon van de letter g



Figuur 3.8 Deel van een rasterbeeldscherm met de geïnverteerde letter g



Figuur 3.6 Deel van een rasterbeeldscherm met de letter g



Figuur 3.7 Deel van een rasterbeeldscherm met de gerooteerde letter g

Een implementatie van het simple DDA algoritme in PASCAL is:

```
procedure dda(x1,y1,x2,y2,c:integer);
  var
    dx,dy,n,i:integer;
    x_increment,y_increment,x,y:real;

  begin
    dx:=x2-x1;
    dy:=y2-y1;
    if abs(dx)>abs(dy)
      then n:=abs(dx)
      else n:=abs(dy);
    x_increment:=dx/n;
    y_increment:=dy/n;
    x:=x1; y:=y1;
    set_pixel(round(x),round(y),c);
    for i:=1 to n do
      begin
        x:=x+x_increment;
        y:=y+y_increment;
        set_pixel(round(x),round(y),c)
      end
    end;
end;
```

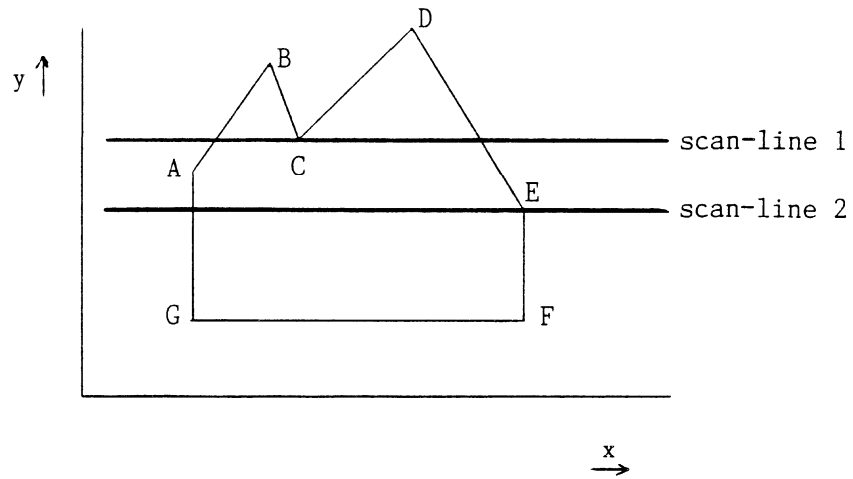
In figuur 3.1 is een gedeelte van een rasterbeeldscherm getekend met een lijnstuk van pixel P(2,1) naar pixel Q(7,4). Een schuine lijn op een rasterbeeldscherm zal een trapsgewijs verloop hebben (aliasing). Door de resolutie te vergroten kan het trapeffect minder zichtbaar worden gemaakt, maar er zijn ook andere anti-aliasing methoden.

Bij de volgende anti-aliasing methode gaat men ervan uit dat een lijn op een beeldscherm een zekere breedte heeft. In figuur 3.2 is tussen de pixels P en Q een lijnstuk met een breedte, gelijk aan die van een pixel, getekend. Het lijnstuk bedekt de omliggende pixels gedeeltelijk. Door de kleur van de pixels af te stemmen op het gedeelte, dat door het lijnstuk wordt bedekt (sampling) kan het trapeffect enigszins gecompenseerd worden (figuur 3.3).

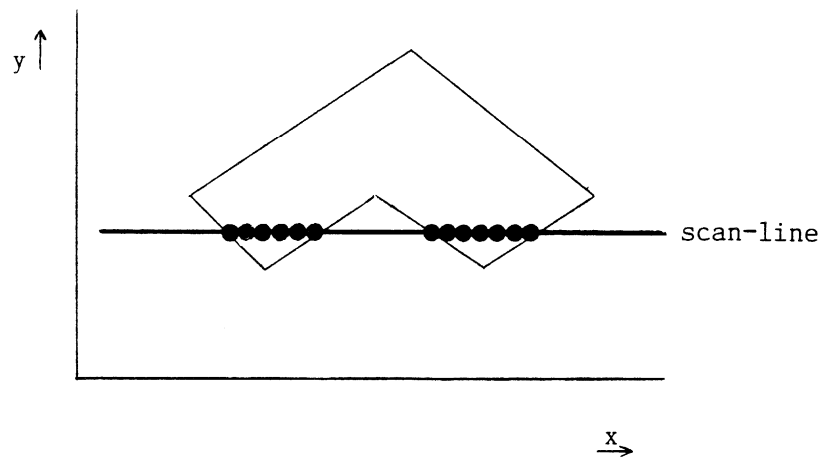
In figuur 3.4 is een horizontaal en een schuin lijnstuk getekend. Beide lijnstukken zijn uit evenveel pixels opgebouwd. Doordat het schuine lijnstuk een factor $\sqrt{2}$ langer is, is het minder helder te zien. Bovenstaande antialiasing methode compenseert ook dit rastereffect.

3.3 Een tekst.

Letters, cijfers en andere tekens kunnen bij een vectorbeeldscherm samengesteld worden m.b.v. korte lijnstukken. Bij een rasterbeeldscherm moeten tekens opgebouwd worden m.b.v. pixels. Patronen voor tekens worden opgeslagen in ROM (Read - Only - Memory).



Figuur 3.10 Een polygoon die door scan-lines doorsneden wordt. De getekende scan-lines gaan door de hoekpunten C en E.



Figuur 3.9 Een polygoon die door een scan-line doorsneden wordt. De pixels op de scan-line binnen de begrenzing zijn getekend.

Figuur 3.5 laat een bitpatroon (mask raster) voor de letter g zien. Om een teken af te beelden op een rasterbeeldscherm (figuur 3.6) wordt het patroon gekopieerd in de framebuffer. De positie van een teken op het scherm wordt bepaald door een translatie van de oorsprong van het bitpatroon naar de gewenste pixel positie. Eventueel kunnen nog andere transformaties, zoals een rotatie (figuur 3.7) worden uitgevoerd. Een teken kan ook geïnverteerd worden afgebeeld (figuur 3.8).

3.4 Een gebied.

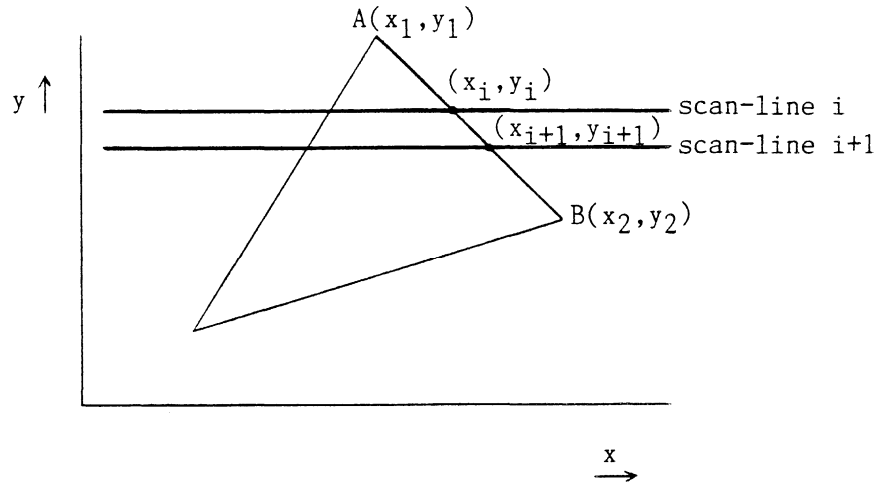
Een gebied wordt m.b.v. een rasterbeeldscherm afgebeeld door de pixels binnen de begrenzing een geschikte kleur te geven. De kleur of het vulpatroon wordt opgeslagen in de framebuffer. De begrenzing kan bijvoorbeeld een cirkel of een polygoon zijn. Een polygoon kan gedefinieerd worden door de opeenvolgende (bijvoorbeeld met de wijzers van de klok mee) hoekpunten. Er zijn verschillende algoritmen ontwikkeld om gebieden op een rasterbeeldscherm met een kleur te vullen. Eén methode gebruikt de geometrische definitie van de begrenzing om te bepalen welke pixels tot het gebied behoren (scan-line algoritme). Een andere methode begint met een pixel binnen het gebied en laat het aantal pixels groeien, totdat de begrenzing die al in de frame buffer aanwezig is, is bereikt (fill-algoritme).

3.4.1 Het scan-line algoritme.

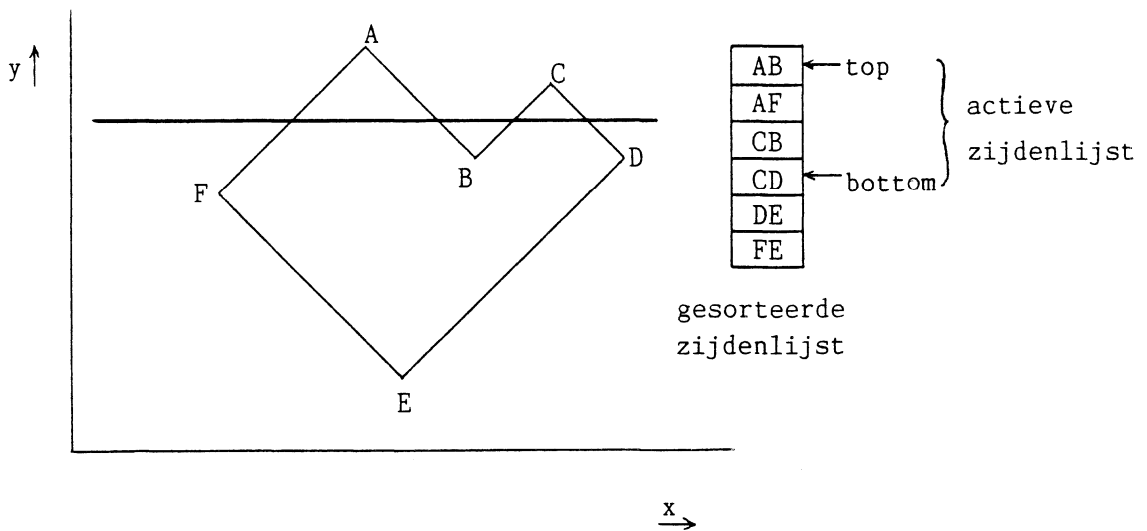
Bij het scan-line algoritme worden voor elke scan-line de snijpunten met de begrenzing van het gebied berekend. Vervolgens wordt bepaald welke pixels binnen het gebied liggen. Figuur 3.9 laat als begrenzing een polygoon zien, dat door een scan-line doorsneden wordt. Van links naar rechts worden de snijpunten paarsgewijs samen genomen waarna de tussenliggende pixels de gewenste kleur krijgen.

Speciale aandacht verdient het geval dat een scan-line door het hoekpunt van een polygoon gaat. Omdat de snijpunten paarsgewijs samen genomen worden, moet het aantal snijpunten steeds even zijn. In figuur 3.10 moet het snijpunt C tweemaal en het snijpunt E éénmaal geteld worden. Hoe vaak een snijpunt meetelt wordt bepaald door de aanliggende zijden. Een snijpunt telt éénmaal mee bij een monotoon niet-stijgend of niet-dalend verloop. In het andere geval, dus als er sprake is van een lokaal maximum of minimum, telt een snijpunt tweemaal mee.

Een scan-conversie algoritme behandelt een polygoon van boven naar beneden en van links naar rechts. De berekeningen die bij zo'n algoritme moeten worden uitgevoerd, kunnen meestal drastisch vereenvoudigd worden door gebruik te maken van de coherentie (samenhang) van de voorwerpen die worden afgebeeld.



Figuur 3.11 Een polygoon die door twee opeenvolgende scan-lines doorsneden wordt.



Figuur 3.12 Een polygoon die door een scan-line doorsneden wordt. De gesorteerde lijst van alle zijden van het polygoon en de met de scan-line corresponderende actieve zijdenlijst zijn aangegeven.

Figuur 3.11 laat twee opeenvolgende scan-lines zien, die één zijde van een polygoon doorsnijden. De zijde wordt door de hoekpunten $A(x_1, y_1)$ en $B(x_2, y_2)$ bepaald. De snijpunten met opeenvolgende scan-lines kunnen m.b.v. de volgende formules berekend worden:

$$x_{i+1} = x_1 - (x_2 - x_1) / (y_2 - y_1)$$

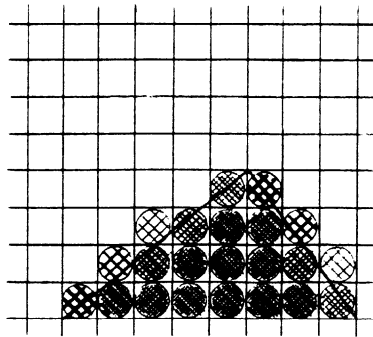
$$y_{i+1} = y_1 - 1.$$

Voor een niet-horizontale zijde kan de x-coördinaat van het snijpunt met een scan-line dus berekend worden door een constante af te trekken van de x-coördinaat van het snijpunt met de voorgaande scan-line. Uitgaande van punt $A(x_1, y_1)$ kunnen de volgende snijpunten m.b.v. de bovenstaande formules worden berekend.

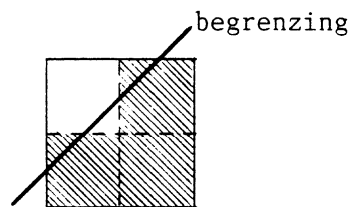
Een bepaalde scan-line zal slechts enkele zijden van een polygoon snijden. Om onnodige controle te voorkomen, kan men een lijst bijhouden met de zijden die door een huidige scan-line doorsneden worden (actieve zijdenlijst). Om dit te realiseren wordt een lijst van alle zijden van het polygoon gemaakt. Deze lijst wordt gesorteerd volgens de y-coördinaat van het hoogste (grootste y-waarde) grenspunt van een zijde. Pointers in deze lijst bepalen de actieve zijdenlijst voor elke scan-line (figuur 3.12). Voordat gestart wordt met het berekenen van de snijpunten met de volgende scan-line, worden de pointers zo nodig aangepast om een nieuwe actieve zijdenlijst te bepalen.

Een gebied kan i.p.v. met een bepaalde kleur ook gevuld worden met een patroon. Het patroon moet opgebouwd worden m.b.v. pixels. Het patroon kan daarom op soortgelijke wijze als in de frame buffer worden vastgelegd. Meestal zal het patroon zich na een klein aantal pixels herhalen, waardoor de benodigde geheugenruimte om het patroon te definiëren beperkt blijft. M.b.v. een translatie kan het definitie-patroon naar de frame buffer gecopiëerd worden.

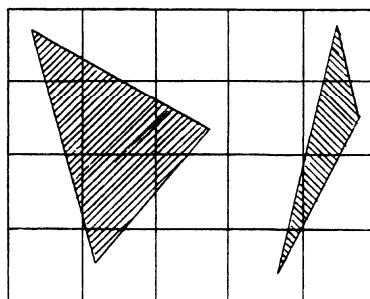
Met een soortgelijke anti-aliasing methode als voor een lijn wordt toegepast, kan het trapsgewijs verloop langs de begrenzing van een gebied gecompenseerd worden (figuur 3.13). De kleur van de pixels wordt nu afgestemd op het gedeelte, dat binnen de begrenzing ligt. Er zijn verschillende methoden ontwikkeld om dit gedeelte te bepalen. Bij één methode wordt een pixel onderverdeeld (subdivision method) in gelijke delen en wordt van elk deel bepaald of het centrum binnen de begrenzing ligt. M.b.v. een scan-line algoritme kan bepaald worden welke delen binnen de begrenzing liggen. In figuur 3.14 is een pixel onderverdeeld in vier delen en ligt 3/4 deel binnen de begrenzing. Door een pixel verder onder te



Figuur 3.13 Deel van een rasterbeeldscherm met een gevuld polygoon, waarbij gebruik is gemaakt van een anti-aliasing methode.



Figuur 3.14 Pixel dat doorsneden wordt door een begrenzing. Het pixel wordt onderverdeeld in vier delen om het gedeelte binnen de begrenzing te bepalen.



Figuur 3.15 Gebieden in een rooster, waarvan de maaswijdte gelijk is aan de breedte van een pixel.

verdelen kan de nauwkeurigheid vergroot worden. De pixel-subdivision methode geeft ook een goede correctie bij een hoekpunt van een polygoon en bij het afbeelden van zéér kleine gebieden (figuur 3.15).

3.4.2 Het boundary-fill algoritme.

Een gebied kan ook met een kleur gevuld worden door met een pixel binnen het gebied te beginnen en verder te gaan met de omliggende pixels, totdat de begrenzing is bereikt. Deze methode, die het boundary-fill algoritme wordt genoemd, is bruikbaar als de begrenzing binnen de framebuffer aanwezig is en een pixel binnen de begrenzing aangewezen kan worden.

De volgende procedure laat een recursieve methode zien om een gebied te vullen m.b.v. de boundary-fill methode, waarbij de gebiedsuitbreiding via de vier horizontaal en verticaal aanliggende pixels plaatsvindt:

```
procedure boundary_fill(x,y,fill_color,boundary_color:integer);
  var
    present_color:integer;
  begin
    present_color:=get_pixel(x,y);
    if (present_color<>boundary_color)
      and (present_color<>fill_color)
    then
      begin
        set_pixel(x,y,fill_color);
        boundary_fill(x+1,y ,fill_color,boundary_color);
        boundary_fill(x ,y+1,fill_color,boundary_color);
        boundary_fill(x-1,y ,fill_color,boundary_color);
        boundary_fill(x ,y-1,fill_color,boundary_color)
      end
    end;
end;
```

De functie get_pixel(x,y) bepaalt de kleur van een pixel in de frame buffer.

3.4.3 Het flood-fill algoritme.

Het flood-fill algoritme lijkt sterk op het boundary-fill algoritme. Ook nu wordt met een pixel binnen het gebied begonnen. De kleur van de omliggende pixels wordt gewijzigd zolang deze kleur hetzelfde is als van het pixel, waarmee begonnen is.

3.5 Prioriteit.

Een plaatje kan zijn opgebouwd uit meerdere gebieden, die elkaar gedeeltelijk overlappen. Als een gebied als één geheel is opgeslagen, bijvoorbeeld door de lijnstukken waaruit de contour is opgebouwd hetzelfde label te geven, kan aan een gebied een bepaalde prioriteit worden toegekend. D.m.v. de prioriteit wordt dan bepaald welk gebied een ander bedekt.

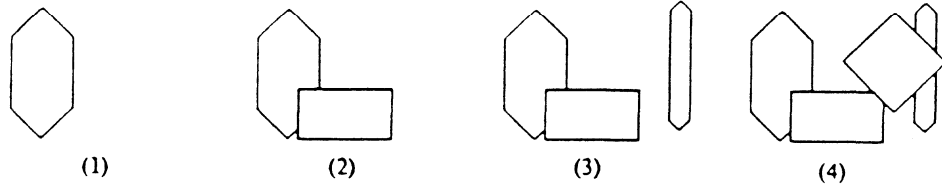
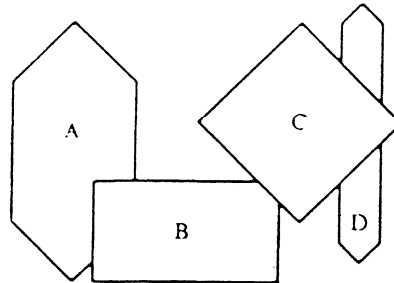
Prioriteit:

A : 1

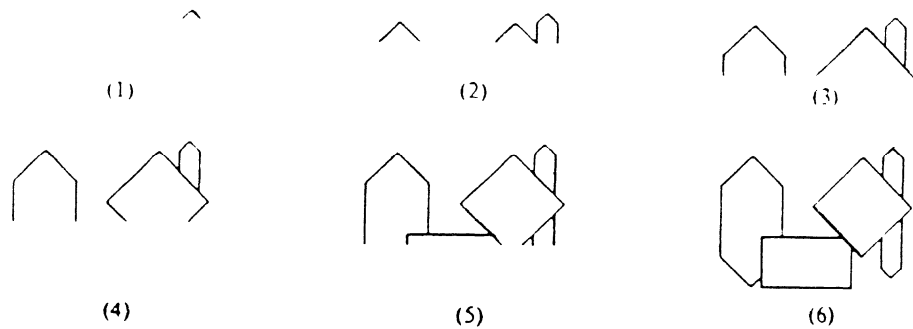
B : 2

C : 4

D : 3



a

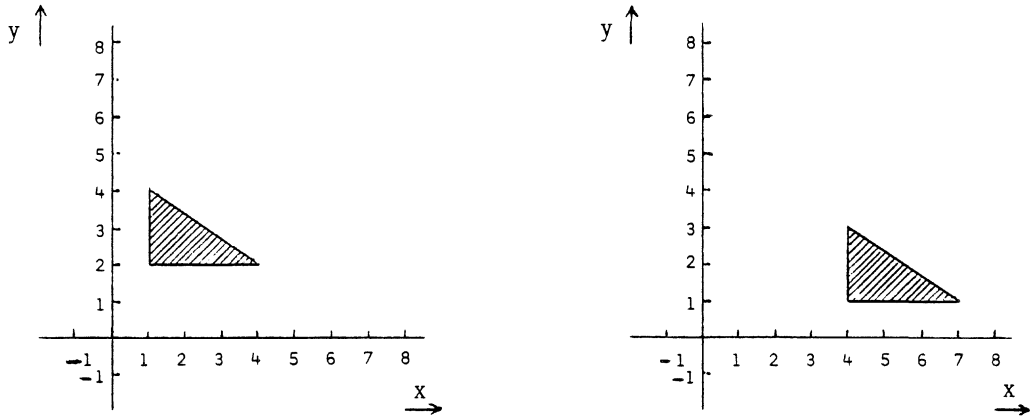


b

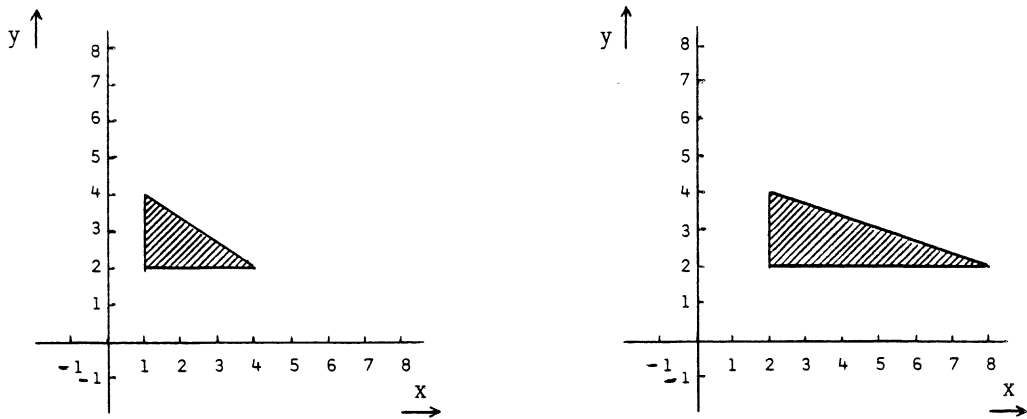
Figuur 3.16 Een plaatje, dat is opgebouwd uit meerdere polygoon, die elkaar gedeeltelijk overlappen. D.m.v. de prioriteit wordt bepaald welk polygoon een ander bedekt. Een plaatje kan worden opgebouwd door polygoon na elkaar op het scherm af te beelden (a). Een andere mogelijkheid is om het plaatje per scan-line van boven naar beneden op te bouwen (b).

Gebieden kunnen na elkaar op het scherm worden afgebeeld (painter's algoritme), waarbij begonnen wordt met het gebied met de laagste prioriteit (figuur 16a). Elk volgend gebied kan het voorgaande bedekken. Bij het painter's algoritme worden voor de scan-conversie alle polygonen gesorteerd volgens de prioriteit.

Een andere mogelijkheid is om tijdens de scan-conversie volgens prioriteit te sorteren. Nadat voor een bepaalde scan-line de snijpunten met de begrenzing van de gebieden zijn berekend, wordt eerst volgens de prioriteit gesorteerd en daarna worden de snijpunten van links naar rechts paarsgewijs samen genomen. Een plaatje wordt hierbij opgebouwd van boven naar beneden (figuur 16b). Dit wordt vaak minder hinderlijk gevonden dan een opbouw volgens het painter's algoritme.



Figuur 4.1 Translatie van een driehoek, waarbij $T_x=3$ en $T_y=-1$.



Figuur 4.2 Schaling van een driehoek t.o.v de oorsprong, waarbij $S_x=2$ en $S_y=1$.

4. Transformaties.

Bij het maken van plaatjes op een beeldscherm wordt veel gebruik gemaakt van transformaties. Om meer details te zien kan een plaatje vergroot worden. Voor een beter overzicht wordt een plaatje juist verkleind. Men kan patronen voor symbolen (standaard figuren) opslaan. Om de symbolen op een bepaalde plaats op het scherm te krijgen wordt een translatie toegepast. Symbolen moeten ook vaak geschaald of geroteerd worden.

In dit hoofdstuk zullen enkele lineaire twee-dimensionale geometrische transformaties worden behandeld. Bij deze transformaties wordt aan een punt $P(x,y)$ een nieuw punt $P'(x',y')$ toegevoegd. Bij het transformeren van een lijnstuk kan dan volstaan worden met de transformatie van de grenspunten, waarna deze rechtstreeks verbonden kunnen worden.

4.1 Basis-transformaties.

4.1.1 Translatie.

Bij een translatie wordt elk punt verschoven over dezelfde afstand in dezelfde richting:

$$\begin{aligned}x' &= x + T_x \\ y' &= y + T_y\end{aligned}\tag{4.1}$$

Het getallenpaar (T_x, T_y) wordt de translatievector genoemd. Figuur 4.1 laat een translatie zien, waarbij $T_x = 3$ en $T_y = -1$.

4.1.2 Schaling.

Bij een schaling t.o.v. de oorsprong worden de coördinaten van elk punt vermenigvuldigd met een getal:

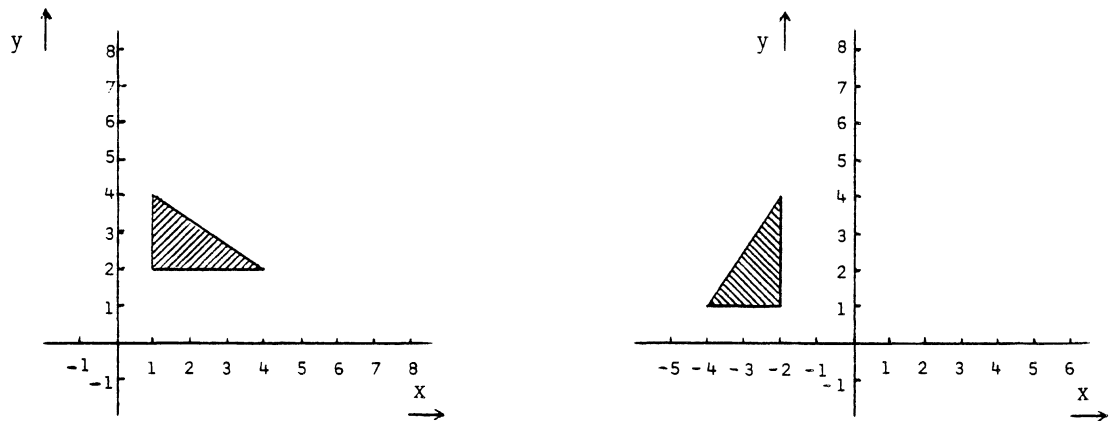
$$\begin{aligned}x' &= x \cdot S_x \\ y' &= y \cdot S_y\end{aligned}\tag{4.2}$$

De oorsprong blijft op zijn plaats (fixed point). S_x en S_y worden de schaalfactoren respectievelijk in de x- en in de y-richting genoemd. Bij een uniforme schaling zijn beide factoren gelijk en treedt er geen vervorming op. Bijzondere gevallen van schaling zijn een lijnspiegeling in de x-as ($S_x = 1$ en $S_y = -1$) en een puntspiegeling t.o.v. de oorsprong ($S_x = -1$ en $S_y = -1$). Figuur 4.2 laat een schaling zien, waarbij $S_x = 2$ en $S_y = 1$.

4.1.3 Rotatie.

Bij een rotatie om de oorsprong wordt elk punt over dezelfde hoek om de oorsprong geroteerd:

$$\begin{aligned}x' &= x \cos\theta - y \sin\theta \\ y' &= x \sin\theta + y \cos\theta\end{aligned}\tag{4.3}$$



Figuur 4.3 Rotatie van een driehoek om de oorsprong over 90° .

De oorsprong is de spil (pivot point) en blijft op zijn plaats. Een positieve waarde voor de rotatiehoek θ betekent een rotatie tegen de wijzers van de klok in. Figuur 4.3 laat een rotatie zien, waarbij $\theta = 90^\circ$.

4.2 Samengestelde transformaties.

Meer algemene transformaties kunnen worden toegepast door het sequentiëel uitvoeren van de basistransformaties. Roteert een punt bijvoorbeeld niet om de oorsprong, maar om een willekeurig ander punt $P(x_p, y_p)$, dan kunnen de transformatie formules worden afgeleid door het achtereenvolgens toepassen van de volgende drie basistransformaties:

- 1) Translatie van P naar de oorsprong:

$$\begin{aligned}x' &= x - x_p \\y' &= y - y_p\end{aligned}$$

- 2) Rotatie om de oorsprong:

$$\begin{aligned}x'' &= x' \cos\theta - y' \sin\theta \\y'' &= x' \sin\theta + y' \cos\theta\end{aligned}$$

- 3) Translatie van de oorsprong naar P:

$$\begin{aligned}x''' &= x'' + x_p \\y''' &= y'' + y_p\end{aligned}$$

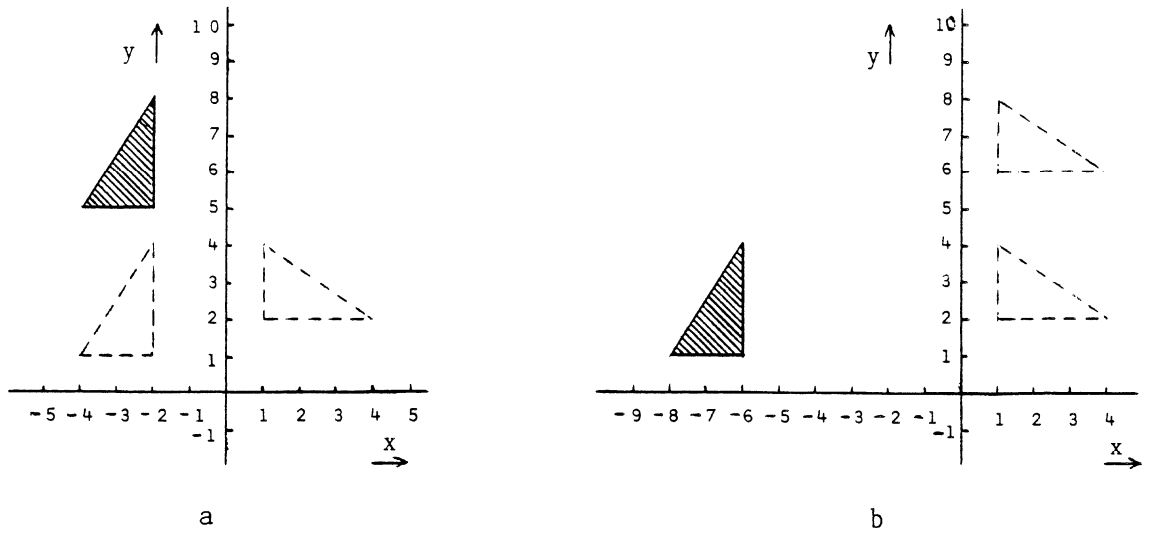
Dus:

$$\begin{aligned}x''' &= x'' + x_p \\&= x' \cos\theta - y' \sin\theta + x_p \\&= (x-x_p) \cos\theta - (y-y_p) \sin\theta + x_p\end{aligned}$$

en

$$\begin{aligned}y''' &= y'' + y_p \\&= x' \sin\theta + y' \cos\theta + y_p \\&= (x-x_p) \sin\theta + (y-y_p) \cos\theta + y_p\end{aligned}$$

Bovenstaande transformaties kunnen samengesteld worden tot één transformatie, waarbij elk punt over dezelfde hoek om punt $P(x_p, y_p)$ roteert:



Figuur 4.4 Een samengestelde transformatie, die uit twee basistransformaties bestaat. In (a) is een driehoek eerst geroteerd om de oorsprong over 90° en vervolgens getransleerd over $(0,4)$. In (b) is de volgorde omgekeerd.

$$\begin{aligned}x' &= (x-x_p)\cos\theta - (y-y_p)\sin\theta + x_p \\y' &= (x-x_p)\sin\theta + (y-y_p)\cos\theta + y_p\end{aligned}\tag{4.4}$$

De volgorde, waarin de transformaties worden uitgevoerd, is in het algemeen bepalend voor het uiteindelijke resultaat. In figuur 4.4a is een driehoek eerst geroteerd om de oorsprong over 90° en vervolgens getransleerd over $(0,4)$. Figuur 4.4b laat het resultaat zien, als de volgorde wordt omgekeerd.

Een voordeel van een samengestelde transformatie is de meer compacte representatie. Ook is het aantal rekenkundige bewerkingen vaak minder dan als de afzonderlijke transformaties sequentiëel worden uitgevoerd.

4.3 Matrix representaties en homogene coördinaten.

De berekening van een samengestelde transformatie wordt aanmerkelijk vereenvoudigd, als gebruik wordt gemaakt van een matrix representatie. Als een punt d.m.v. homogene coördinaten wordt aangegeven kunnen de formules voor de basistransformaties (ook de translatie) in een overeenkomstige matrix vorm worden geschreven. In plaats van door het getallenpaar (x,y) wordt een punt in homogene coördinaten gerepresenteerd door het drietal $[x_h, y_h, w]$, waarbij:

$$\begin{aligned}x_h &= x \cdot w \\y_h &= y \cdot w\end{aligned}\tag{4.5}$$

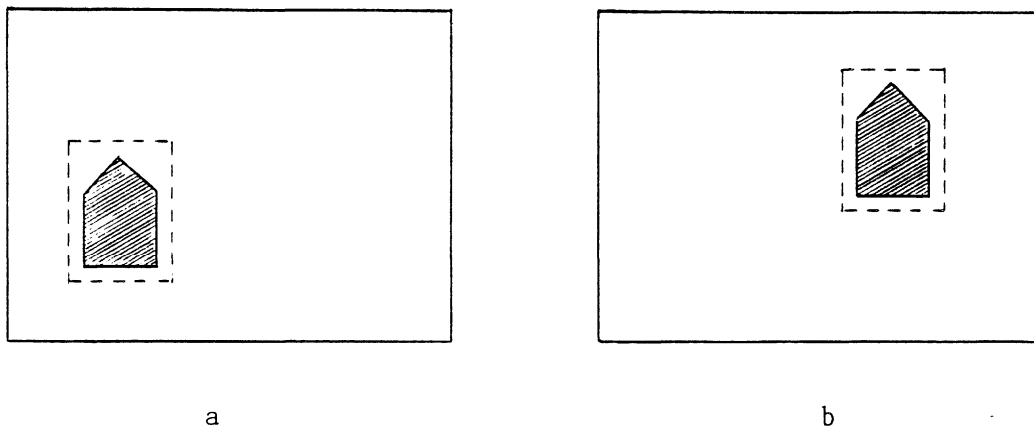
Voor de besproken transformaties kan w gelijk worden gesteld aan 1. De basistransformaties kunnen dan als volgt in matrix vorm worden geschreven:

$$\text{Translatie: } [x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}$$

$$\text{Schaling: } [x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rotatie: } [x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Samengestelde transformaties kunnen direct opgeschreven en berekend worden door het achtereenvolgens vullen van de afzonderlijke transformatie matrices. Bijvoorbeeld voor de voorgaande transformatie, waarbij elk punt over dezelfde hoek om punt $P(x_p, y_p)$ roteert, geldt:



Figuur 4.5 Translatie binnen de frame buffer (bit-block transfer). De pixels binnen een rechthoekig gebied worden gekopieerd naar het nieuwe deel. Daarna wordt binnen het oorspronkelijke deel de achtergrondkleur ingesteld.

$$\begin{aligned}
 [x' \ y' \ 1] &= [x \ y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_p & -y_p & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_p & y_p & 1 \end{bmatrix} \\
 &= [x \ y \ 1] \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ -x_p \cos\theta + y_p \sin\theta + x_p & -x_p \sin\theta - y_p \cos\theta + y_p & 1 \end{bmatrix} \quad (4.6)
 \end{aligned}$$

De basis- en samengestelde transformaties zijn allen te schrijven in de vorm:

$$[x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} a & d & 0 \\ b & e & 0 \\ c & f & 1 \end{bmatrix} \quad (4.7)$$

Hierin zijn a t/m f constanten, die voor een bepaalde transformatie slechts één keer berekend moeten worden. Voor de berekening van de coördinaten moeten dan nog voor elk punt 4 vermenigvuldigingen en 4 optellingen worden uitgevoerd:

$$\begin{aligned}
 x' &= ax + by + c \\
 y' &= dx + ey + f
 \end{aligned} \quad (4.8)$$

4.4 Transformaties binnen de framebuffer.

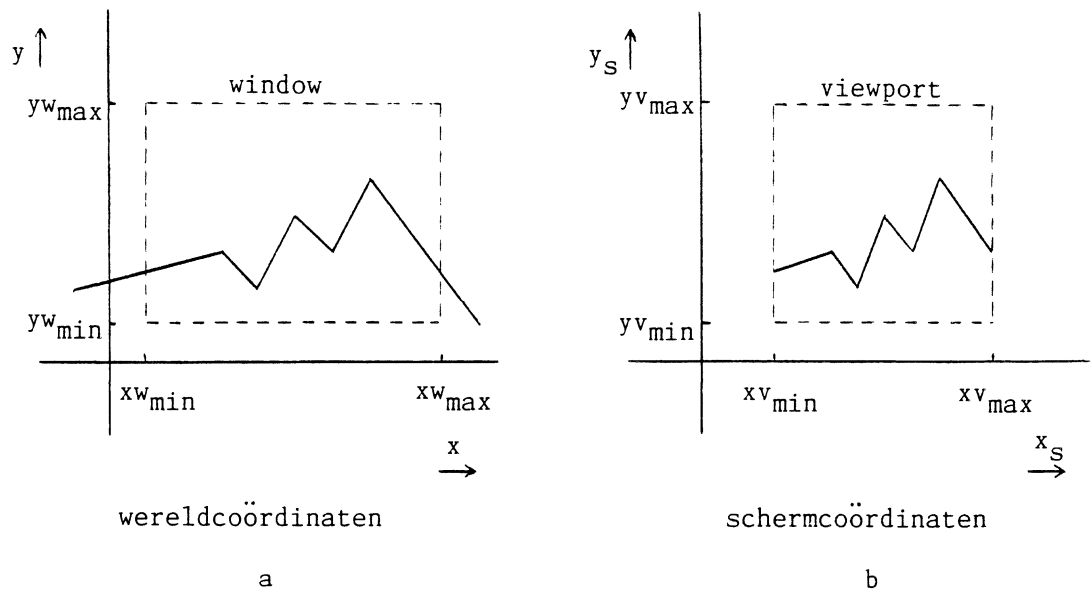
Een frame buffer is niet alleen een hulpmiddel om een plaatje op een rasterbeeldscherm af te beelden, maar bevat ook een representatie van het plaatje. Een paar eenvoudige transformaties kunnen direct binnen de framebuffer worden uitgevoerd, waardoor een aanmerkelijke snelheidswinst kan worden behaald.

Figuur 4.5 laat een translatie zien, waarbij een rechthoekig gedeelte van de afbeelding op het scherm wordt verplaatst. Daartoe moeten de pixels in het oorspronkelijke deel naar het nieuwe deel gecopiëerd worden. Daarna kan binnen het oorspronkelijke deel de achtergrondkleur worden ingesteld. Als het oorspronkelijke deel het nieuwe overlapt, moet wel de juiste volgorde bij het kopiëren in acht worden genomen. Een pixel in het oorspronkelijke deel mag namelijk pas veranderd worden als het gecopiëerd is.

Behalve een translatie van een rechthoekig gedeelte kan ook een rotatie over 90° worden uitgevoerd. Daartoe worden rijen (scan-lines) en kolommen (pixels) verwisseld:

4.5 Window en viewport.

De posities van punten op een kaart, tekening, enz. worden meestal vastgelegd t.o.v. een coördinatensysteem, dat voor de betreffende toepassing het meest geschikt is. De cartesische coördinaten van een punt t.o.v. dit coördinatensysteem noemt



Figuur 4.6 Transformatie van de wereldcoördinaten naar de schermcoördinaten (viewing transformation).

men de wereldcoördinaten. Wil men een plaatje afbeelden op een beeldscherm, dan moet er een transformatie plaatsvinden van de wereldcoördinaten naar de schermcoördinaten (viewing transformation).

Figuur 4.6 laat zien hoe het een en ander in zijn werk gaat. Het window (in wereldcoördinaten) bepaalt het rechthoekige gebied in de wereldruimte dat moet worden afgebeeld. Het viewport (in schermcoördinaten) bepaalt het rechthoekige gebied op het scherm waar de afbeelding moet plaatsvinden. Een punt $P(x, y)$ binnen het window wordt getransformeerd naar een punt $P'(x_s, y_s)$ binnen het viewport, waarbij geldt:

$$x_s = \frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}} (x - x_{wmin}) + x_{vmin} \quad (4.9)$$

$$y_s = \frac{y_{vmax} - y_{vmin}}{y_{wmax} - y_{wmin}} (y - y_{wmin}) + y_{vmin}$$

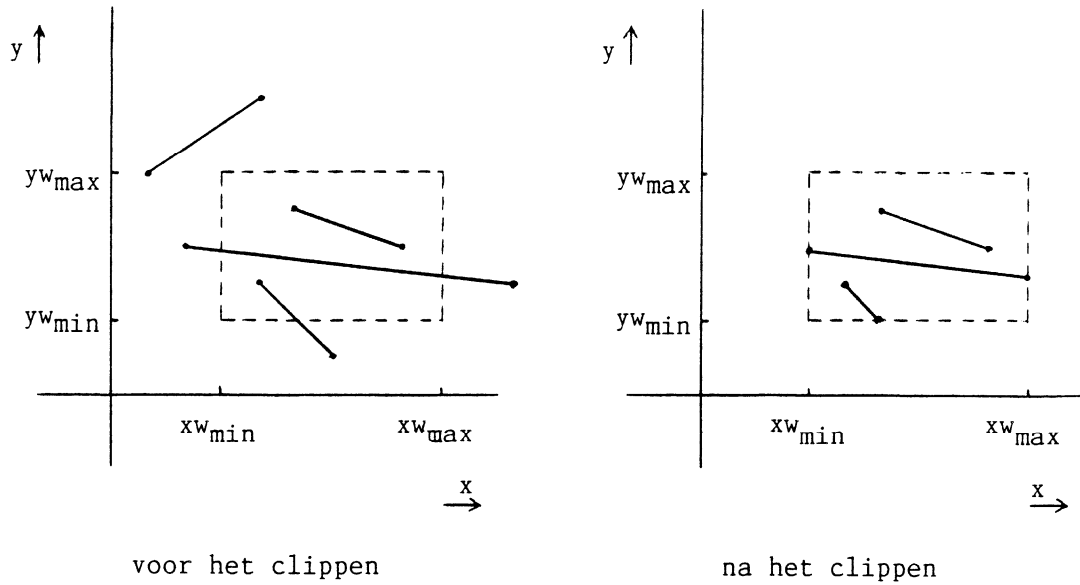
Deze transformatie kan opgevat worden als een transformatie, die is samengesteld uit een translatie, een schaling en een translatie. Wil er geen vervorming optreden, dan moeten de schaalfactoren in de x- en y-richting gelijk zijn, dus:

$$\frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}} = \frac{y_{vmax} - y_{vmin}}{y_{wmax} - y_{wmin}} \quad (4.10)$$

De schermcoördinaten zijn afhankelijk van het beeldscherm. Als men verschillende uitvoerapparaten gebruikt, kan men beter werken met normalized device coördinates (NDC), waarvan de waarden tussen 0.0 en 1.0 liggen. In dat geval moet er, voordat een plaatje m.b.v. een bepaald uitvoerapparaat wordt afgebeeld, nog een transformatie naar de coördinaten van het betreffende apparaat worden uitgevoerd.

4.6 Clipping.

Van een plaatje wordt alleen het gedeelte, dat zich binnen het window bevindt, afgebeeld. Het weglaten van de gedeelten van een plaatje buiten het window wordt clipping genoemd. Het "clippen" kan zowel tegen de grenzen van het window als tegen de grenzen van het viewport gebeuren. Vooral als men maar een klein gedeelte van een plaatje wil zien, en dus een groot deel zich buiten het window bevindt, is het efficiënter om tegen de grenzen van het window te clippen. Er hoeven dan minder punten getransformeerd te worden. In het volgende zal er van worden



Figuur 4.7 Het clippen van enkele lijnstukken tegen de grenzen van het window.

{left,top}	{top}	{right,top}
{left}	{ }	{right}
{left,bottom}	{bottom}	{right,bottom}

Figuur 4.8 De in 9 zones verdeelde wereldruimte. De ligging van een grenspunt wordt m.b.v. een verzameling aangegeven.

uitgegaan, dat er geclippt wordt tegen de grenzen van het window.

4.6.1 Clipping van een punt.

Het clippen van een punt betekent dat er bepaald moet worden of een punt buiten de grenzen van het window ligt. Een punt $P(x,y)$ ligt binnen of op de grens van het window, als de coördinaten van het punt aan de volgende vier ongelijkheden voldoen:

$$x_{wmin} \leq x \leq x_{wmax} \wedge y_{wmin} \leq y \leq y_{wmax} \quad (4.11)$$

Wordt niet aan alle ongelijkheden voldaan, dan ligt een punt buiten het window en moet het worden weggelaten.

4.6.2 Clipping van een lijn.

Het clippen van een lijn betekent dat bepaald moet worden welk deel van een lijnstuk buiten de grenzen van het window ligt. Van lijnen die gedeeltelijk buiten de begrenzing liggen, moeten de snijpunten met de begrenzing berekend worden. Aangezien een plaatje uit duizenden lijnstukken kan bestaan, moet de nodige aandacht aan de efficiëntie van het clipproces worden besteed.

Figuur 4.7 laat de ligging van enkele lijnstukken t.o.v. de grenzen van het window zien. De ligging van een lijnstuk t.o.v. de grenzen van het window kan bepaald worden d.m.v. de ligging van de grenspunten:

- Als beide grenspunten binnen het window liggen, ligt het hele lijnstuk binnen het window.
- Als het ene grenspunt binnen en het andere buiten het window ligt is er één snijpunt. Het gedeelte buiten het window moet worden weggelaten.
- Als beide grenspunten buiten het window liggen zijn er twee mogelijkheden:
 1. Er is geen snijpunt. Het hele lijnstuk ligt buiten het window en moet worden weggelaten.
 2. Er zijn twee (eventueel samenvallende) snijpunten. De gedeelten buiten het window moeten worden weggelaten.

Cohen en Sutherland hebben een algoritme ontwikkeld, dat eerst de ligging van een lijnstuk t.o.v. de grenzen van het window bepaalt en vervolgens, als een lijnstuk een grens snijdt, het snijpunt berekent en het buitenliggende deel weglaat. Door de grenzen van het window te verlengen, wordt de wereldruimte in 9 zones verdeeld. De ligging van een grenspunt wordt m.b.v. een verzameling aangegeven (figuur 4.8). De mogelijke elementen van deze verzameling zijn:

left d.w.z. het grenspunt ligt links van de lijn $x = x_{wmin}$
right d.w.z. het grenspunt ligt rechts van de lijn $x = x_{wmin}$
bottom d.w.z. het grenspunt ligt onder de lijn $y = y_{wmin}$
top d.w.z. het grenspunt ligt boven de lijn $y = y_{wmax}$

Een lijnstuk ligt geheel binnen het window als van beide grenspunten de verzameling leeg is. Een lijnstuk ligt geheel buiten het window als de doorsnede van deze verzamelingen niet leeg is. In alle andere gevallen snijdt het lijnstuk één of meer (verlengde) grenzen van het window en wordt het snijpunt berekend.

Als bijvoorbeeld de grenspunten van een lijnstuk $P(x_1, y_1)$ en $Q(x_2, y_2)$ zijn en als P links van de lijn $x = x_{wmin}$ ligt (Q ligt dan niet links van deze lijn) wordt het snijpunt met de lijn $x = x_{wmin}$ bepaald:

$$x = x_{wmin}$$
$$y = \frac{y_2 - y_1}{x_2 - x_1} (x_{wmin} - x_1) + y_1$$

Het lijnstuk links van de lijn $x = x_{wmin}$ wordt weggelaten door het grenspunt P te vervangen door het snijpunt. Vervolgens wordt de code van het nieuwe grenspunt bepaald.

Het voorgaande proces wordt herhaald, totdat een lijnstuk geheel binnen of geheel buiten het window ligt.

Een implementatie van het Cohen en Sutherland algoritme in PASCAL is:

```
var
  xw_min, xw_max, yw_min, yw_max: real;

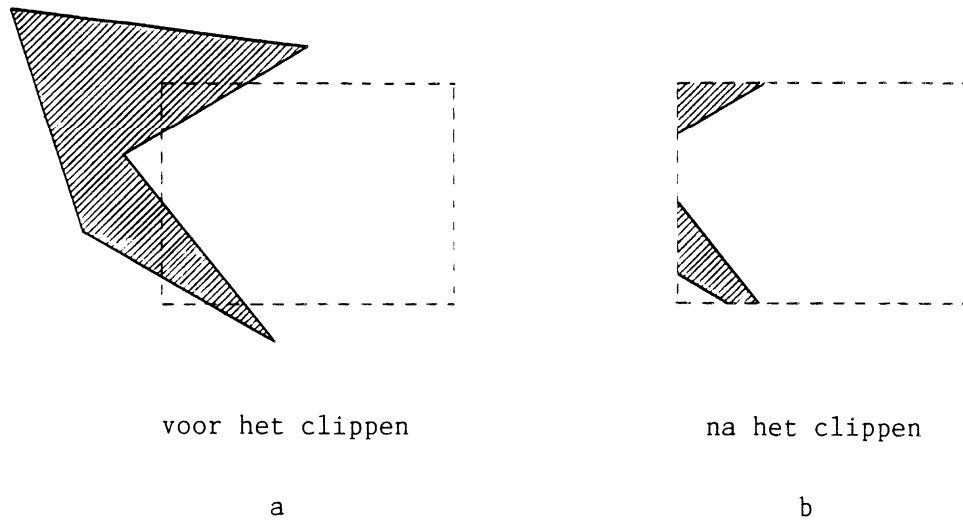
procedure clip_a_line(x1, y1, x2, y2: real);
  type
    edge = (left, right, bottom, top);
    code = set of edge;
  var
    c, c1, c2: code;
    x, y: real;

  procedure encode(x, y: real; var c: code);
    begin
      c := [];
      if x < xw_min
      then c := [left]
      else if x > xw_max then c := [right];
```

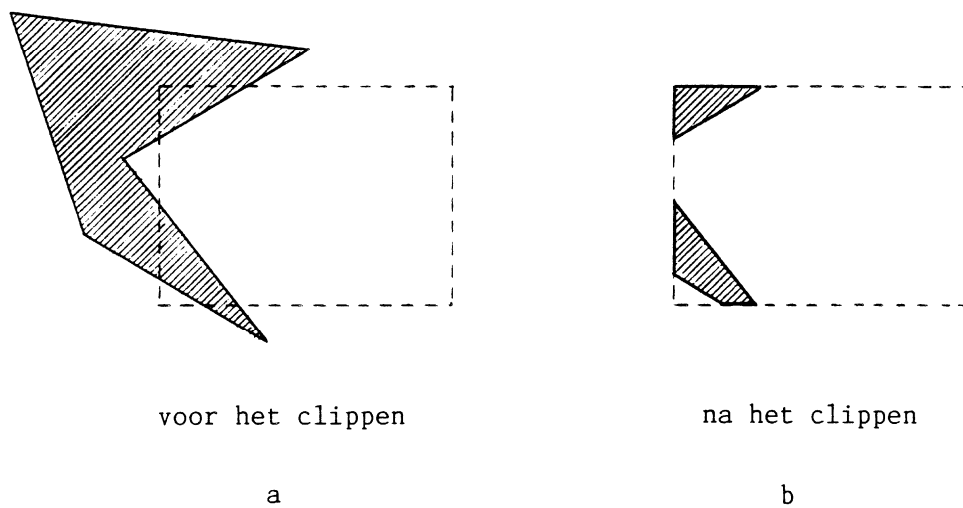


```
    if y<yw_min
      then c:=c+[bottom]
    else if y>yw_max then c:=c+[top]
  end; {encode}

begin {clip_a_line}
  encode(x1,y1,c1); encode(x2,y2,c2);
  while ((c1<>[]) or (c2<>[])) and ((c1*c2)=[]) do
    begin
      if c1<>[]
        then c:=c1
        else c:=c2;
      if left in c
        then
          begin
            x:=xw_min;
            y:=(y2-y1)/(x2-x1)*(xw_min-x1)+y1
          end
        else
          if right in c
            then
              begin
                x:=xw_max;
                y:=(y2-y1)/(x2-x1)*(xw_max-x1)+y1
              end
            else
              if bottom in c
                then
                  begin
                    x:=(x2-x1)/(y2-y1)*(yw_min-y1)+x1;
                    y:=yw_min
                  end
                else
                  if top in c
                    then
                      begin
                        x:=(x2-x1)/(y2-y1)*(yw_max-y1)+x1;
                        y:=yw_max
                      end;
            end;
      if c1<>[]
        then
          begin
            x1:=x; y1:=y;
            encode(x1,y1,c1)
          end
        else
          begin
            x2:=x; y2:=y;
            encode(x2,y2,c2)
          end
      end; {while}
    if (c1=[]) and (c2=[])
      then {teken de lijn van (x1,y1) naar (x2,y2)}
    end; {clip_a_line}
```



Figuur 4.9 Het clippen van een polygoon tegen de grenzen van het window d.m.v. een line-clipping algoritme.



Figuur 4.10 Het clippen van een polygoon tegen de grenzen van het window d.m.v. een polygoon-clipping algoritme.

4.6.3 Clipping van een polygoon.

In figuur 4.9a wordt een gebied begrensd door een polygoon. Figuur 4.9b laat het resultaat zien als de zijden van het polygoon één voor één geclippt worden m.b.v. een line-clipping algoritme. Het zichtbare gebied is niet meer omsloten. Als de zijden van een polygoon een gebied begrenzen, dat gevuld moet worden, moet het line-clipping algoritme worden aangepast, zodat het gebied omsloten blijft (figuur 4.10).

Sutherland en Hodgman hebben een algoritme ontwikkeld, waarbij de zijden van het polygoon tegen de vier verlengde grenzen van het window worden geclippt. Het polygoon wordt gedefiniëerd door de opeenvolgende hoekpunten. Er zijn dan vier mogelijkheden:

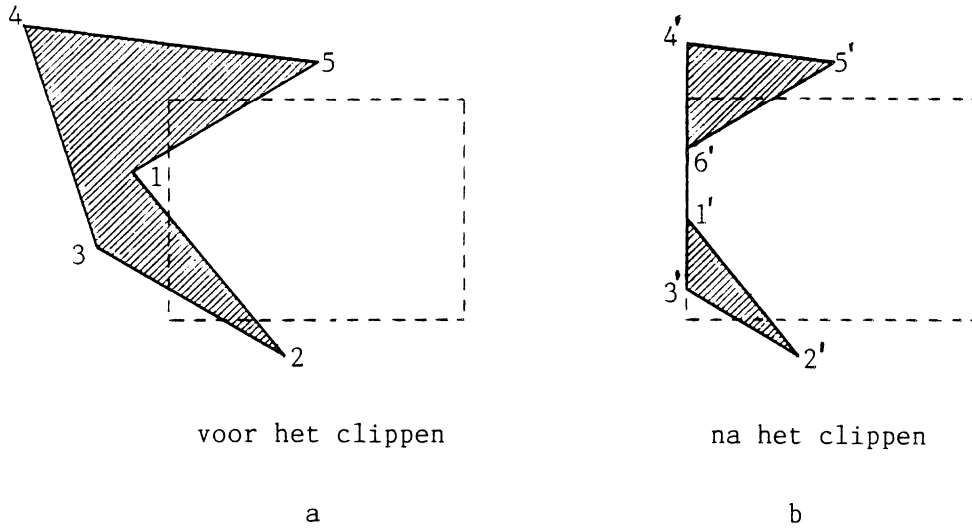
1. Het beginpunt en het eindpunt liggen aan de buitenzijde:
 - Het eindpunt is geen hoekpunt van het geclippte polygoon.
2. Het beginpunt ligt aan de buitenzijde. Het eindpunt niet.
 - Het snijpunt is een hoekpunt van het geclippte polygoon. (Een stuk van de windowgrens moet in de geclippte polygoon worden opgenomen)
 - Het eindpunt is een hoekpunt van het geclippte polygoon.
3. Het eindpunt ligt aan de buitenzijde. Het beginpunt niet.
 - Het snijpunt is een hoekpunt van het geclippte polygoon.
 - Het eindpunt is geen hoekpunt van het geclippte polygoon.
4. Het beginpunt en het eindpunt liggen niet aan de buitenzijde:
 - Het eindpunt is een hoekpunt van het geclippte polygoon.

Figuur 4.11 laat een polygoon zien, dat tegen de linkergrens van het window is geclippt. De hoekpunten van het geclippte polygoon zijn met een accent aangegeven. Als het hele polygoon eerst tegen één grens wordt geclippt en daarna tegen de volgende grens, moeten alle hoekpunten tussentijds worden opgeslagen. Als een zijde van het polygoon tegen de vier grenzen wordt geclippt en daarna de volgende zijde, hoeven er tussentijds minder hoekpunten te worden opgeslagen. Alleen bij mogelijkheid 2 is het tussentijds opslaan van een hoekpunt noodzakelijk.

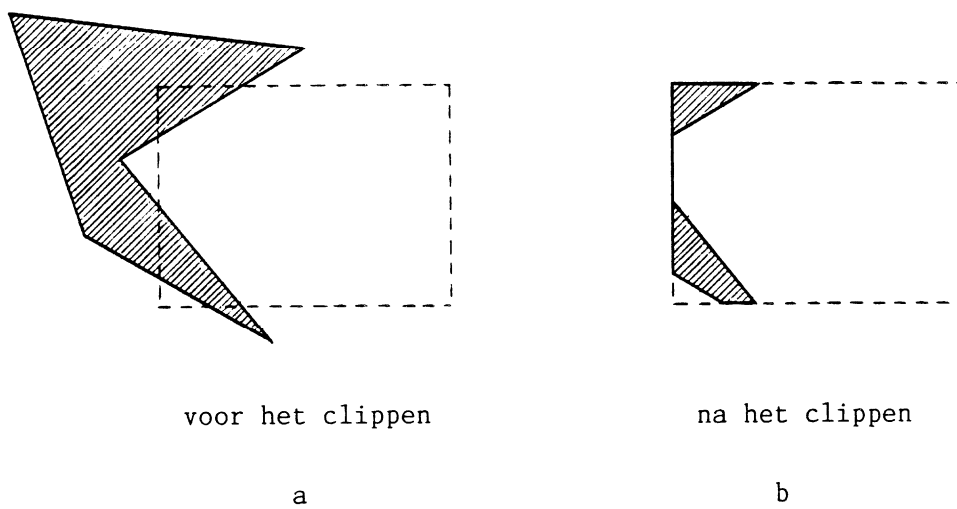
Als een concaaf polygoon wordt geclippt, kan het zichtbare gebied uit twee of meer afzonderlijke delen bestaan. Het geclippte polygoon wordt gedefiniëerd door één opsomming van opeenvolgende hoekpunten, waardoor de afzonderlijke delen door grenslijnen verbonden zullen zijn (figuur 4.12).

4.6.4 Clipping van een tekst.

Een eenvoudige en snelle methode om een tekst te clippen is het weglaten van de tekens, die niet geheel zichtbaar zijn. Daartoe kan een rechthoekige box om het teken worden aangebracht. Alleen als alle hoekpunten van de box binnen het zichtbare gebied liggen, wordt het teken afgebeeld. Het is ook mogelijk om de afzonderlijke delen van een teken te



Figuur 4.11 Het clippen van een polygoon tegen de linkergrens van het window d.m.v. het algoritme van Sutherland en Hodgman.



Figuur 4.12 Het clippen van een polygoon tegen de grenzen van het window d.m.v. het algoritme van Sutherland en Hodgman.

clippen. Als een teken d.m.v. lijnstukken is gedefiniëerd kan een line-clipping algoritme worden toegepast. Is een teken gedefiniëerd d.m.v. een bitpatroon, dan kunnen de afzonderlijke pixels (punten) worden geclippt (zie 4.6.1).

5. Invoertechnieken.

5.1 Logische classificatie van invoerapparatuur.

Bij grafische toepassingsprogramma's kunnen verschillende soorten gegevens worden ingevoerd. Bijvoorbeeld de coördinaten van een punt, een tekst of getallen voor een transformatie. Wil men een deel van een tekening verwijderen, dan zullen de betreffende elementen aangegeven moeten worden, bijvoorbeeld door ze aan te wijzen. Vaak zal er een keuze gemaakt moeten worden uit een menu.

In samenhang met de verschillende soorten invoergegevens kan er een logische classificatie van invoerapparatuur (logical device classification) worden gemaakt:

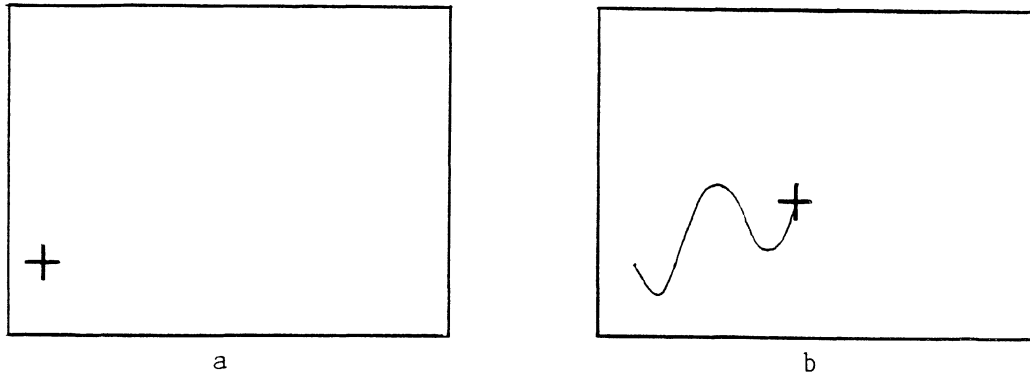
LOCATOR - voor het invoeren van de coördinaten van een punt.
STROKE - voor het invoeren van de coördinaten van een serie punten.
STRING - voor het invoeren van een tekst.
VALUATOR - voor het invoeren van een reëel getal.
PICK - voor het aanwijzen van elementen op het scherm.
CHOICE - voor het invoeren van een geheel getal, b.v. bij het kiezen uit een menu.

Voor hetzelfde logische invoerapparaat b.v. LOCATOR device kunnen meerdere fysieke invoerapparaten b.v. een muis of een tablet worden gebruikt. Omgekeerd kan hetzelfde fysieke invoerapparaat b.v. een muis voor verschillende logische invoerapparaten b.v. een LOCATOR of een PICK device worden gebruikt. Wel zal het ene apparaat voor het invoeren van bepaalde gegevens meer geschikt zijn dan het andere. Door gebruik te maken van logische invoerapparaten, die naar een bepaald soort gegevens vragen kan een programma apparaatonafhankelijk worden gemaakt. Als men in een toepassingsprogramma gegevens invoert, moet het logische apparaat door het fysieke apparaat en de bijbehorende programmatuur (apparaat-afhankelijk) gesimuleerd worden.

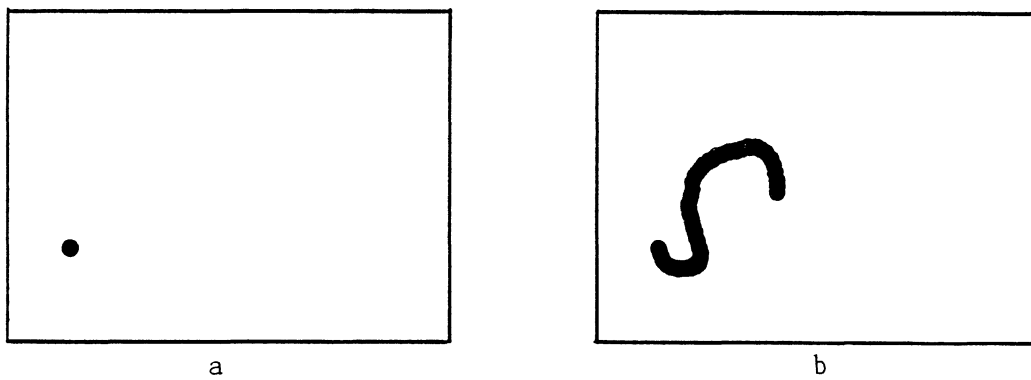
5.1.1 Het invoeren van de coördinaten van een punt (LOCATOR).

Coördinaten kunnen worden ingevoerd m.b.v. een toetsenbord door de getallen in te typen (grote nauwkeurigheid). Als men direct op het scherm tekent zijn de coördinaten echter vaak niet bekend. Bij het direct op het scherm tekenen is het handig, als men m.b.v. een aantal toetsen een cursor op het scherm kan verplaatsen. Op veel toetsenborden zitten vier speciale toetsen om de cursor naar links, naar rechts, naar beneden of naar boven te verplaatsen. Een cursor kan ook verplaatst worden m.b.v. een regelpaneel, een joystick of een muis. Als de cursor op de gewenste plaats op het scherm is, moet één of andere toets worden ingedrukt om de coördinaten van dat punt vast te leggen. De nauwkeurigheid van de ingevoerde coördinaten is afhankelijk van de resolutie van het beeldscherm en de vaste hand van de gebruiker.

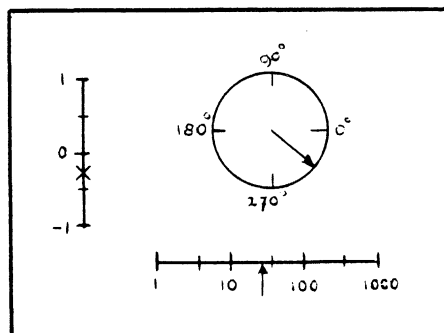
Heeft men een tablet dan kan m.b.v. een handcursor of een stylus de cursor direct naar de juiste positie worden



Figuur 5.1 Inking op het beeldscherm. Tijdens het verplaatsen van de cursor wordt er meestal een toets ingedrukt gehouden.



Figuur 5.2 Painting op het beeldscherm. Tijdens het verplaatsen van de cursor wordt er meestal een toets ingedrukt gehouden.



Figuur 5.3 Het aanwijzen van getallen op een schaal, die op het scherm is afgebeeld.

gebracht (absolute besturing). M.b.v. een tablet kan men ook de coördinaten van punten op een tekening invoeren. Door drie karakteristieke punten (die niet op een rechte lijn liggen) op het tablet aan te wijzen, wordt het coördinatenstelsel vastgelegd. Vervolgens kan men de coördinaten van alle punten bepalen (digitaliseren).

5.1.2 Het invoeren van de coördinaten van een serie punten (STROKE).

De posities van de punten worden meestal, terwijl er een toets ingedrukt gehouden wordt, met zeer korte tussenpozen ingevoerd. Als de posities op het scherm d.m.v. puntjes worden afgebeeld, ontstaat er een spoor van punten, dat de verplaatsing van het invoerapparaat aangeeft. Worden de punten door lijnstukjes verbonden, dan lijkt het resultaat veel op het schrijven op papier (figuur 5.1). Deze techniek wordt daarom wel inking genoemd. M.b.v. een rasterbeeldscherm kunnen ook op eenvoudige wijze de breedte van de lijn en de kleurschakering worden ingesteld. Het resultaat lijkt dan veel op het schilderen met een penseel (figuur 5.2). Men spreekt in dit geval wel van painting.

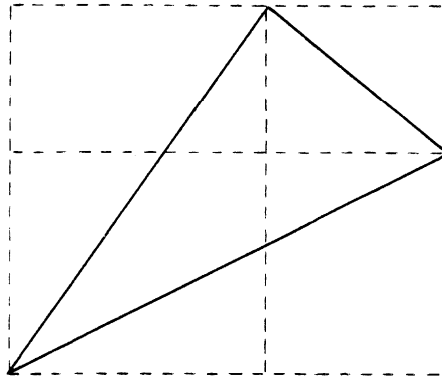
Apparaten, die geschikt zijn om de coördinaten van een enkel punt in te voeren zijn meestal ook wel te gebruiken voor het invoeren van een serie coördinaten. Vanwege de bestuurbaarheid is echter niet elk apparaat even geschikt. Een tablet is door de absolute besturing heel geschikt voor inking en painting, maar ook een muis is meestal goed bruikbaar.

5.1.3 Het invoeren van een tekst (STRING).

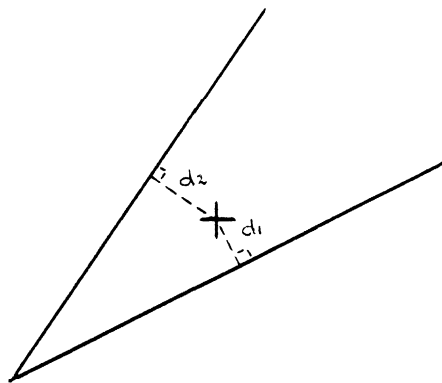
Teksten kunnen m.b.v. een toetsenbord worden ingevoerd door het typen van de afzonderlijke tekens. Veel gebruikte teksten kunnen eventueel m.b.v. een enkele functietoets worden ingevoerd.

5.1.4 Het invoeren van een getal (VALUATOR).

Getallen kunnen m.b.v. een toetsenbord worden ingevoerd. Ook een regelpaneel, een joystick, een muis of een tablet kunnen gebruikt worden om getallen in te voeren, bijvoorbeeld door de grootte van het getal afhankelijk te maken van de verplaatsing of verdraaiing. Een andere mogelijkheid is het aanwijzen van getallen op een schaal die op het scherm is afgebeeld (figuur 5.3). D.m.v. een cursor, bijvoorbeeld in de vorm van een kruis of een pijl, kan een getal op de schaal worden aangewezen. Hierbij kan ook van een lichtpen gebruik worden gemaakt. M.b.v. een logaritmische schaal kan over het hele interval dezelfde relatieve nauwkeurigheid worden bereikt. D.m.v. een goede terugkoppeling, bijvoorbeeld door het aangewezen getal ergens op het scherm af te beelden kan de gebruiker meer zekerheid t.a.v. de grootte worden gegeven.



Figuur 5.4 Het aanwijzen van elementen op het beeldscherm. Om elk element kan een rechthoek worden aangebracht. Binnen een rechthoekig gebied kunnen meerdere elementen liggen.



Figuur 5.5 Het aanwijzen van elementen op het beeldscherm. De kortste afstand bepaald het aangewezen element.

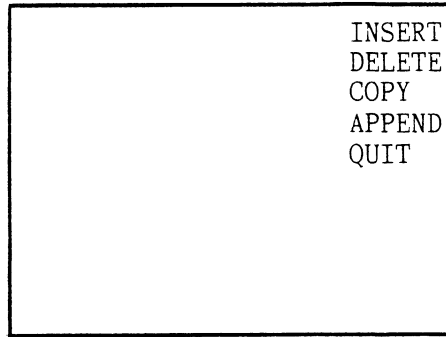
5.1.5 Het aanwijzen van elementen op het scherm (PICK).

M.b.v. een lichtpen kan een element op het scherm worden aangewezen. Als tijdens de refresh cycle de elektronenstraal de plaats van de pen bereikt en de knop op de pen ingedrukt is, zal het element bepaald worden. Bij een vectorbeeldscherm kan het element, dat op dat moment wordt opgefrist, direct in de displayfile worden opgezocht. Bij een rasterbeeldscherm zal m.b.v. de framebuffer bepaald moeten worden met welk element het aangewezen pixel correspondeert. Soms kunnen er meerdere stappen nodig zijn om het juiste element te vinden. Eerst kan er een rechthoek om elk element worden angebracht (figuur 5.4). Als het aangewezen pixel binnen het rechthoekige gebied van één element ligt is het aangewezen element gevonden. Als het pixel binnen meerdere gebieden ligt kan vervolgens de afstand tot elk element worden bepaald (figuur 5.5). In dat geval is het element dat het dichtstbij ligt het aangewezen element.

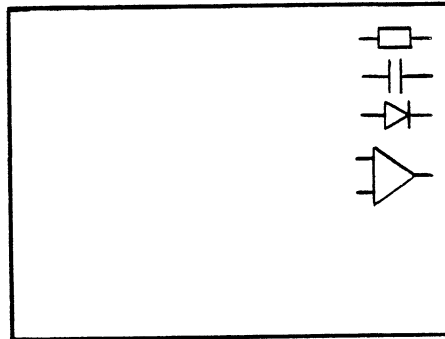
Met een cursor of een kruisdraad kan ook m.b.v. andere invoerapparaten, zoals een functietoetsenbord, een joystick, een muis of een tablet, een element op het scherm worden aangewezen. De cursor moet dan zo dicht mogelijk in de buurt van het beoogde element worden gebracht. Met de coördinaten van de aangewezen plaats kan m.b.v. de bovengenoemde techniek, zowel bij het vectorbeeldscherm als bij het rasterbeeldscherm, bepaald worden welk element wordt aangewezen.

Als een gebruiker de cursor naar een hoekpunt van een polygoon brengt, wijst hij dan naar het hoekpunt of naar een zijde? Soms wil een gebruiker niet één element, maar meerdere tegelijk aanwijzen. De elementen vormen bijvoorbeeld samen een voorwerp dat hij wil verplaatsen. Er kunnen verschillende technieken worden toegepast om een gebruiker een eenduidige keuze te laten maken:

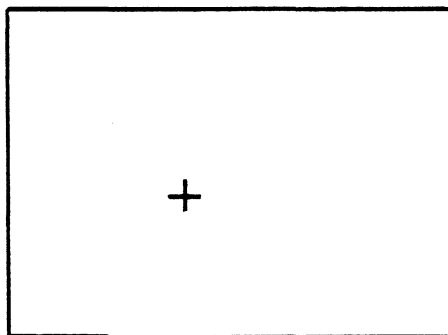
- Een grafisch element kan aangewezen worden door naar een speciale plaats te wijzen, zoals het middelpunt van een cirkel of het eindpunt van een lijn. De betekenis van een dergelijk punt kan met een symbool worden aangegeven, bijvoorbeeld c = cirkel en e = eindpunt. Om de gebruiker te helpen kunnen deze punten benadrukt worden door een hogere intensiteit of door knipperen.
- Het beoogde element kan aangewezen worden door er een rechthoek of een cirkel omheen te plaatsen. Deze techniek is vooral nuttig als er meerdere elementen gekozen moeten worden.
- Door gebruik te maken van verschillende toetsen op het invoerapparaat kan de gebruiker aangeven welk soort element er wordt gewenst. Dus de ene toets geeft een punt aan, de volgende een lijn, enz. De verschillende toetsen op een muis en een handcursor zijn hiervoor goed te gebruiken.
- Door direct voorafgaand een opdracht te geven, waarmee duidelijk wordt gemaakt om welk soort element het gaat. M.b.v. een dergelijke opdracht kan ook aangegeven worden wat er met het aangegeven element moet gebeuren,



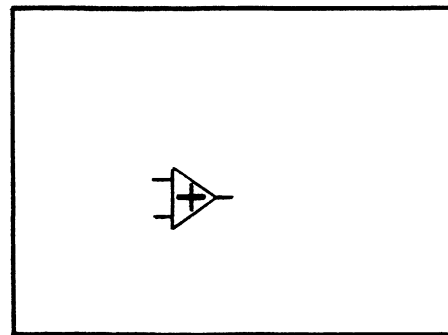
Figuur 5.6 Het kiezen uit een menu. De verschillende mogelijkheden zijn langs de rechterzijde van het beeldscherm aangegeven.



Figuur 5.7 Het kiezen uit een menu. De verschillende elementen, waaruit gekozen kan worden, zijn langs de rechterzijde van het beeldscherm aangegeven.



a



b

Figuur 5.8 Het plaatsen van een symbool.

bijvoorbeeld `delete_point`, `delete_line`, `delete_circle`, enz.

- Door een bepaalde gebruiksmode in te stellen, waarbij alleen met een bepaald soort element wordt gewerkt. Deze techniek biedt vooral voordelen, als er meerdere handelingen met dezelfde soort elementen moeten worden uitgevoerd.

Bovengenoemde technieken zijn slechts enkele van de vele, die mogelijk zijn. Welke techniek toegepast moet worden is afhankelijk van hetgeen geselecteerd moet worden en hoe vaak dit moet gebeuren. Is voor een bepaalde toepassing éénmaal voor een bepaalde techniek gekozen, dan moet deze zo consequent mogelijk worden doorgevoerd.

Een gebruiker moet niet alleen duidelijk gemaakt worden wat hij kan aanwijzen en hoe, maar ook wat hij aangewezen heeft. Een goede terugkoppeling (feedback) kan gegeven worden door hetgeen aangewezen is:

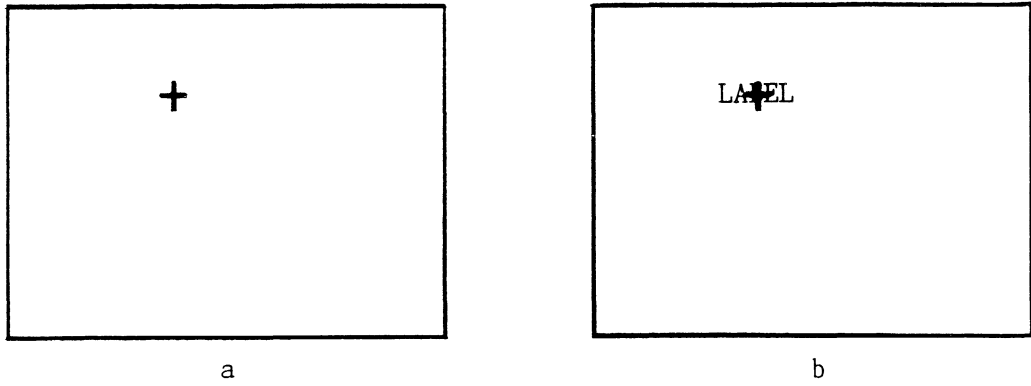
- een hogere intensiteit te geven
- te laten knipperen
- te markeren d.m.v. een rechthoek of een cirkel
- in een andere lijnsoort te tekenen, bijvoorbeeld met een streepjeslijn i.p.v. een doorgetrokken lijn
- een andere kleur te geven.

Als de gebruiker het beoogde element heeft gevonden kan hij de selectie afronden d.m.v. het indrukken van een toets.

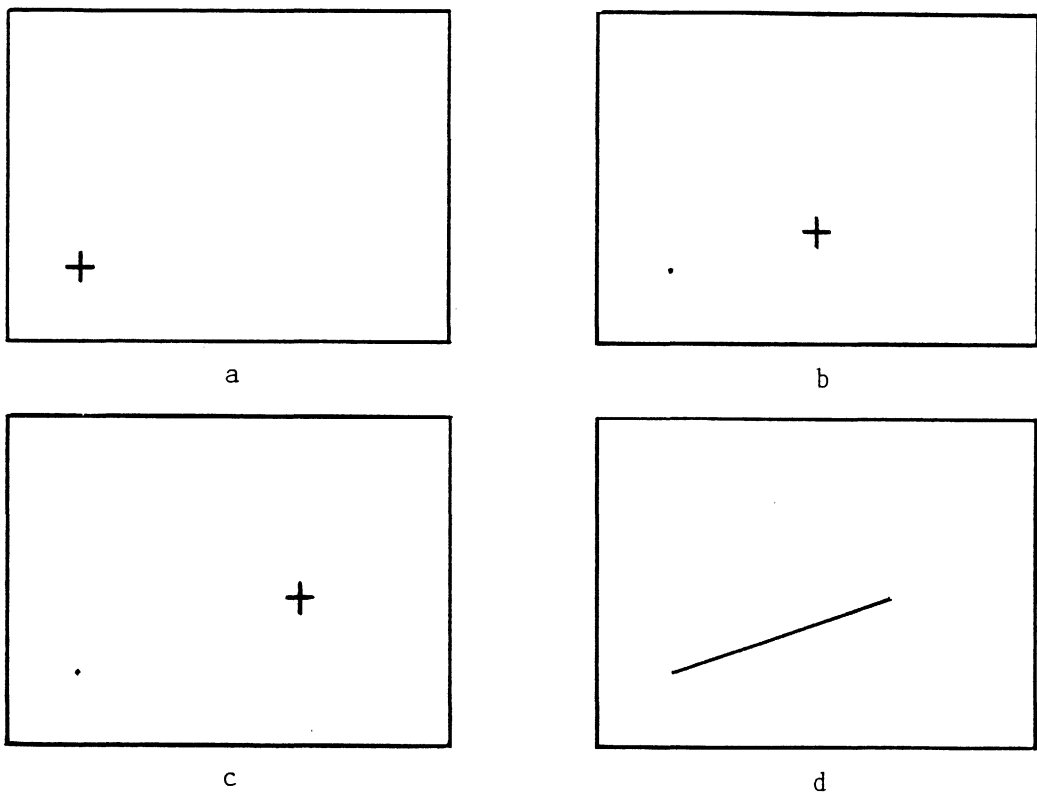
5.1.6 Het kiezen uit een menu (CHOICE).

Door het afbeelden van een menu op het scherm kan men een gebruiker een goed inzicht geven t.a.v. de mogelijkheden van een programma. Door een duidelijk menu kan voorkomen worden dat de gebruiker een programma verkeerd gebruikt. Een menu wordt vaak langs één van de verticale zijden van het scherm afgebeeld (figuur 5.6). Om een menu overzichtelijk te laten zijn, moeten de mogelijkheden waaruit gekozen kan worden, m.b.v. een korte tekst of een symbool (figuur 5.7) aangegeven worden. Vooral bij tekenprogramma's kan een menu met bijvoorbeeld tekenelementen en arceringen zeer efficiënt zijn. Bij veel keuzemogelijkheden kan beter gebruik worden gemaakt van submenu's. Als de menu's logisch en consequent zijn opgebouwd zal de gebruiker al na korte tijd zijn weg binnen het programma kunnen vinden.

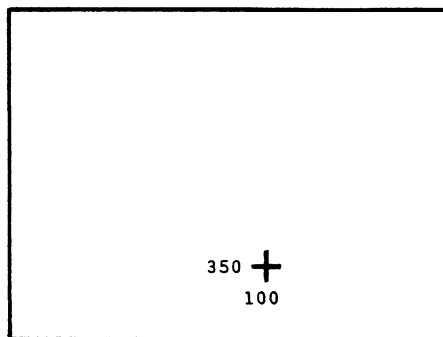
Vaak kan m.b.v. een toetsenbord een keuze uit het menu worden gemaakt door een getal of een letter te typen of door het indrukken van een functietoets. Bij veel menu's zijn de verschillende keuzemogelijkheden binnen rechthoekige velden geplaatst. Door de cursor binnen zo'n veld te brengen kan een bepaalde keuzemogelijkheid worden aangewezen. Hiervoor kunnen de in de vorige paragraaf genoemde invoerapparaten worden gebruikt. D.m.v. het inverteren van het aangewezen veld of één van de in de vorige paragraaf genoemde



Figuur 5.9 Het plaatsen van een tekst.



Figuur 5.10 Het tekenen van een lijn.



Figuur 5.11 Het bepalen van een plaats, terwijl de coördinaten naast de cursor worden afgebeeld.

terugkoppelingstechnieken kan de gebruiker meer zekerheid t.a.v. de potentiële keuze worden gegeven. D.m.v. het indrukken van een toets kan de keuze definitief worden gemaakt. Heeft men de beschikking over een tablet, dan zal bij grafische programma's ook vaak een keuze gemaakt kunnen worden uit een menu dat is vastgemaakt op het tablet.

5.2 Tekentechnieken

5.2.1 Plaats bepalen (positioning).

Een plaats kan bepaald worden door de coördinaten m.b.v. een toetsenbord in te voeren. Een plaats kan ook op het scherm worden aangewezen door er een cursor naar toe te brengen.

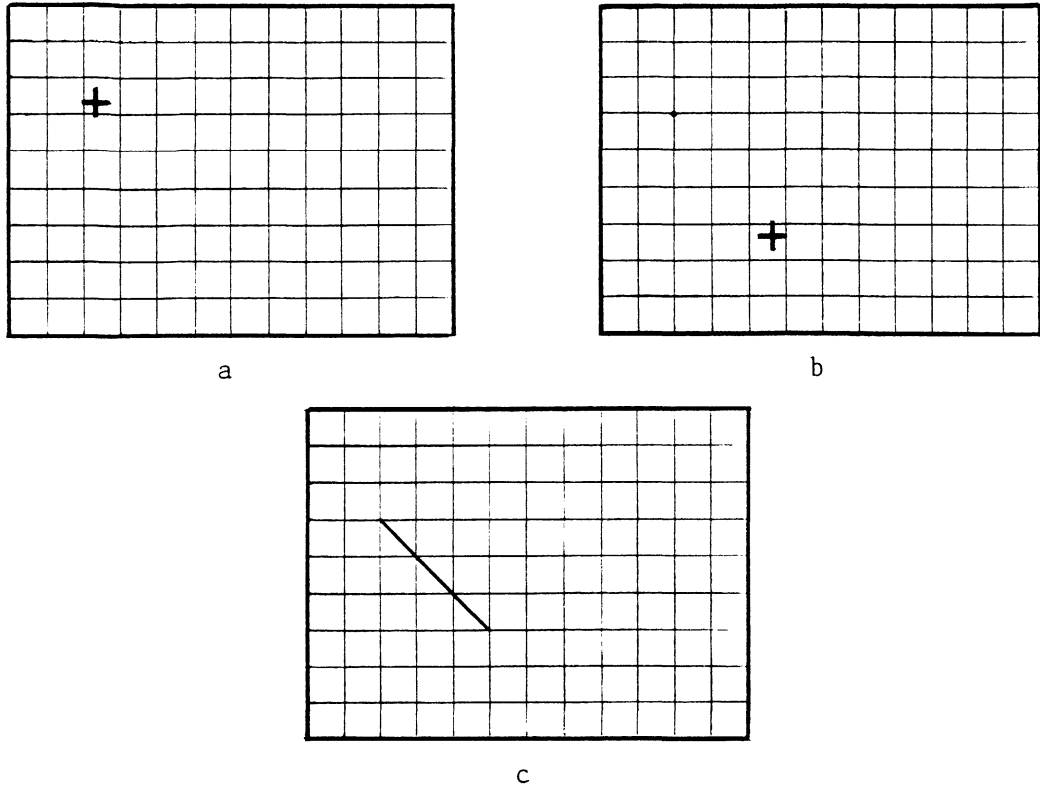
Figuur 5.8 laat zien hoe een symbool op een bepaalde plaats wordt getekend. De cursor wordt naar de gewenste plaats gebracht, vervolgens wordt een toets ingedrukt om aan te geven dat het symbool op deze plaats moet worden afgebeeld. Op een soortgelijke wijze kan de plaats van een tekst bepaald worden. Voor een nauwkeurige plaatsbepaling moet de gebruiker weten waar de oorsprong van het symbool of de tekst ligt. Bij een symbool ligt de oorsprong meestal in een markant punt, bijvoorbeeld een symmetriecentrum. Voor een tekst kan de oorsprong bijvoorbeeld de linkeronderhoek, het midden of de rechterbovenhoek zijn. Figuur 5.9 laat zien hoe de plaats van een tekst wordt bepaald als de oorsprong in het midden ligt.

Figuur 5.10 laat zien hoe een lijn wordt getekend. Eerst wordt de plaats van het beginpunt bepaald en daarna de plaats van het eindpunt. Vervolgens wordt een lijn tussen deze twee punten getekend. Ook andere geometrische figuren, zoals rechthoeken en cirkels kunnen zo, door de plaats te bepalen van enkele markante punten, getekend worden.

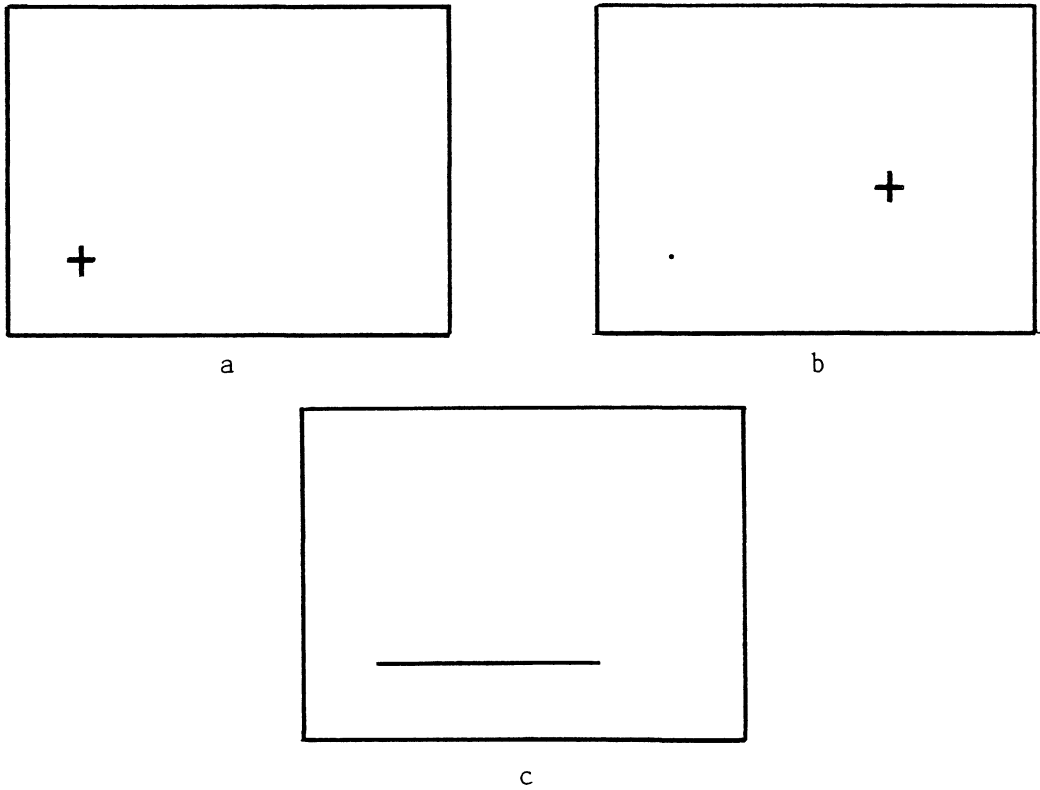
Om een plaats nauwkeuriger te kunnen aanwijzen, kunnen de coördinaten van de cursor worden aangegeven op het scherm. Figuur 5.11 laat zien hoe de coördinaten naast de cursor worden afgebeeld.

5.2.2 Beperkingen t.a.v. plaats en richting (modular and directional constraints).

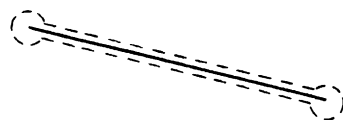
Stel dat men bijvoorbeeld voor een stroomdiagram een aantal symbolen precies onder of naast elkaar op het scherm wil tekenen. Zelfs als men een vaste hand heeft, is dit zonder speciale hulp van de computer geen eenvoudig karwei. Als het aantal mogelijke plaatsen beperkt wordt tot de punten van een rooster (grid), zal het eenvoudiger zijn om de symbolen op de juiste plaats te krijgen. De afstand tussen de roosterpunten moet zo groot zijn, dat het onderscheid tussen twee roosterpunten goed te zien is. De afstand tussen de roosterpunten mag ook niet te groot zijn, omdat anders het aantal mogelijke posities te klein wordt. De afstand tussen de roosterpunten kan meestal door de gebruiker worden ingesteld. Ook voor het onder elkaar zetten van enkele regels



Figuur 5.12 Het tekenen van een lijn m.b.v. een modulair constraint.



Figuur 5.13 Het tekenen van een lijn m.b.v. een directional constraint.



Figuur 5.14 De attractiesfeer rondom een lijn.

tekst (linker kantlijn en regelafstand) en voor het tekenen van bepaalde figuren, bijvoorbeeld een rechthoek, kan een rooster een nuttig hulpmiddel zijn. Als een willekeurige plaats op het scherm wordt aangewezen, zullen door het programma de coördinaten van het dichtstbij gelegen roosterpunt worden genomen. Figuur 5.12 laat het tekenen van een lijn zien. Soms kunnen alleen de roosterpunten worden aangewezen en springt de cursor of kruisdraad tijdens het verplaatsen van het ene naar het andere roosterpunt. Voor de goede werking hoeft het rooster niet afgebeeld te worden. In sommige gevallen zal het zichtbaar zijn van het rooster een nuttige visuele steun geven, terwijl in andere gevallen het juist als hinderlijk ervaren wordt. Ook de zichtbaarheid van het rooster zal daarom meestal door de gebruiker ingesteld kunnen worden. Behalve roosters kunnen er ook andere beperkingen t.a.v. de plaats toegepast worden, zoals het alleen kunnen invoeren van punten die op een cirkel liggen.

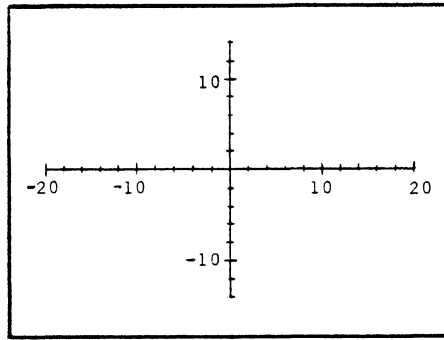
Voor het tekenen van evenwijdige lijnen kan het nuttig zijn, als het aantal mogelijke richtingen beperkt wordt. In veel toepassingen b.v. netwerken, worden alleen horizontale en verticale lijnen gebruikt. In figuur 5.13 wijst de gebruiker twee punten aan. Het programma tekent vervolgens een horizontale of verticale lijn, afhankelijk van welke lijn de twee punten het dichtst benadert.

5.2.3 Attractiesfeer (gravity-field).

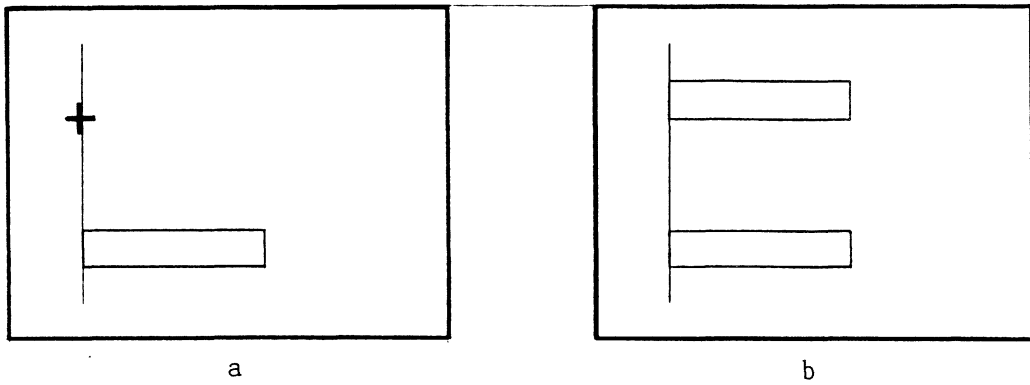
Bij het tekenen van figuren zal een gebruiker vaak een nieuwe lijn willen vastmaken aan een al eerder getekende lijn. Als er met een rooster wordt gewerkt, kan er van de roosterpunten gebruik worden gemaakt. Soms zal een gebruiker echter een lijn op een andere plaats willen vastmaken. Daar het vrijwel onmogelijk is om een cursor precies op een punt van een lijn te plaatsen, zijn er programma's gemaakt, die, als een punt in de buurt van een lijn wordt aangewezen, het dichtstbij gelegen punt op de lijn nemen. In de buurt van de lijn bevindt zich een gebied, waarbinnen een punt als het ware door de lijn wordt aangetrokken. In figuur 5.14 is deze attractiesfeer met de vorm van een halter aangegeven. In de buurt van de grenspunten is het gebied groter gemaakt, zodat een gebruiker daar gemakkelijker lijnen vast kan maken. De grootte van de attractiesfeer moet zodanig zijn dat een gebruiker een nieuwe lijn gemakkelijk vast kan maken. De attractiesfeer mag echter ook niet te groot zijn, omdat anders de gebieden van verschillende lijnen elkaar teveel overlappen, waardoor het moeilijker wordt om de punten juist aan te geven. Gewoonlijk worden de grenzen van de attractiesfeer niet afgebeeld, en kiest de gebruiker eenvoudig punten die zo dicht mogelijk in de buurt van de lijn liggen.

5.2.4 Rooster, schaal en hulplijn.

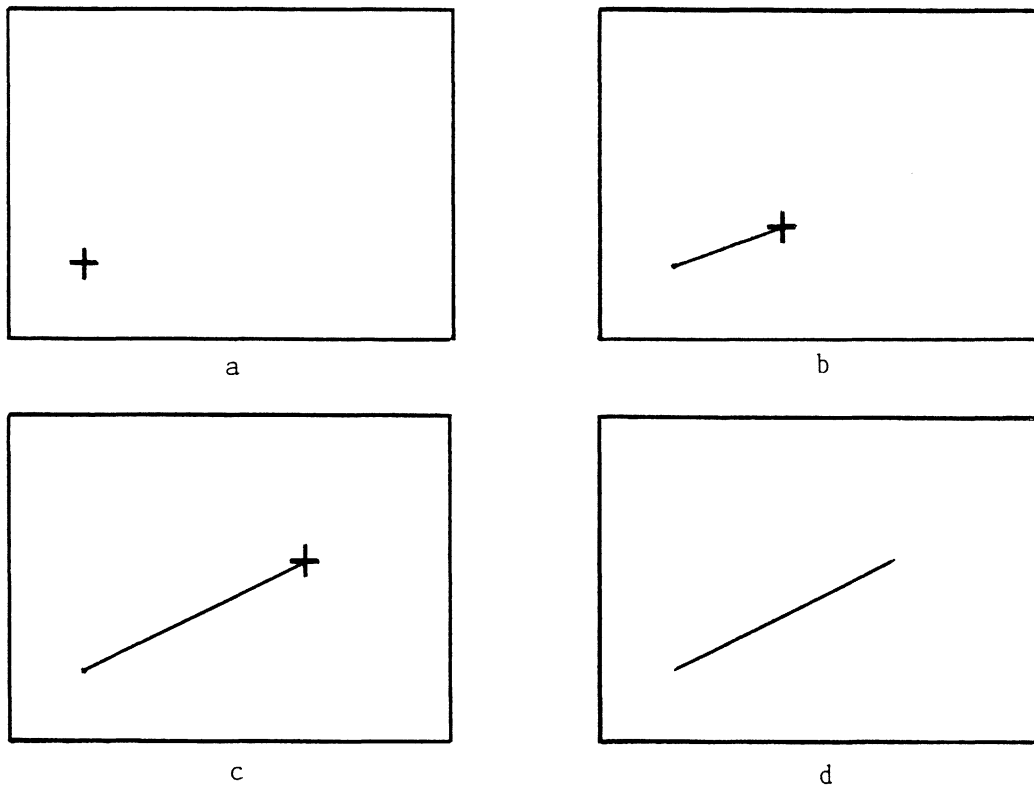
Om een lengte of een hoek beter te kunnen schatten, wordt soms gebruik gemaakt van een rooster zonder dat er beperkingen zijn opgelegd t.a.v. de plaats die kan worden aangewezen of ingevoerd. I.p.v. een rooster wordt ook wel



Figuur 5.15 Het gebruik maken van een schaal.



Figuur 5.16 Het gebruik maken van een hulplijn.



Figuur 5.17 Het tekenen van een lijn m.b.v. de rubber-band techniek.

gebruik gemaakt van een schaal, die dan langs één van de assen (of beide) wordt afgebeeld (figuur 5.15). Omdat een schaal ook hinderlijk kan zijn, wordt het meestal aan de gebruiker overgelaten of hij de schaal al of niet zichtbaar wil maken.

Met een hulplijn is het soms eenvoudiger om een hoek aan te geven of voorwerpen precies onder elkaar te plaatsen (figuur 5.16). Het gebruik van een schaal en een hulplijn is vooral nuttig als het afbeelden van een heel rooster hinderlijk gevonden wordt.

5.2.5 Rubber-band techniek.

Figuur 5.17 laat zien hoe een lijn m.b.v. de rubber-band techniek wordt getekend. Als de gebruiker het beginpunt heeft ingevoerd en de cursor naar het eindpunt brengt, wordt een lijn afgebeeld die het eerste punt met de cursor verbindt. De gebruiker kan dus de ligging van de lijn zien, voordat deze definitief getekend wordt. Als de cursor over het scherm wordt verplaatst, lijkt het of deze d.m.v. elastiek met het beginpunt verbonden is. M.b.v. de rubber-band techniek wordt het eenvoudig om een lijn door of juist langs andere elementen te tekenen.

Ook andere geometrische figuren kunnen op deze wijze getekend worden, bijvoorbeeld een rechthoek (figuur 5.18) of een cirkelboog (figuur 5.19). Vooral bij complexere figuren zijn rubber-band technieken nuttig, omdat het daar moeilijker is om vooraf aan de hand van invoerpunten een beeld van het resultaat te krijgen (hierbij worden wel hoge eisen aan de verwerkingssnelheid gesteld).

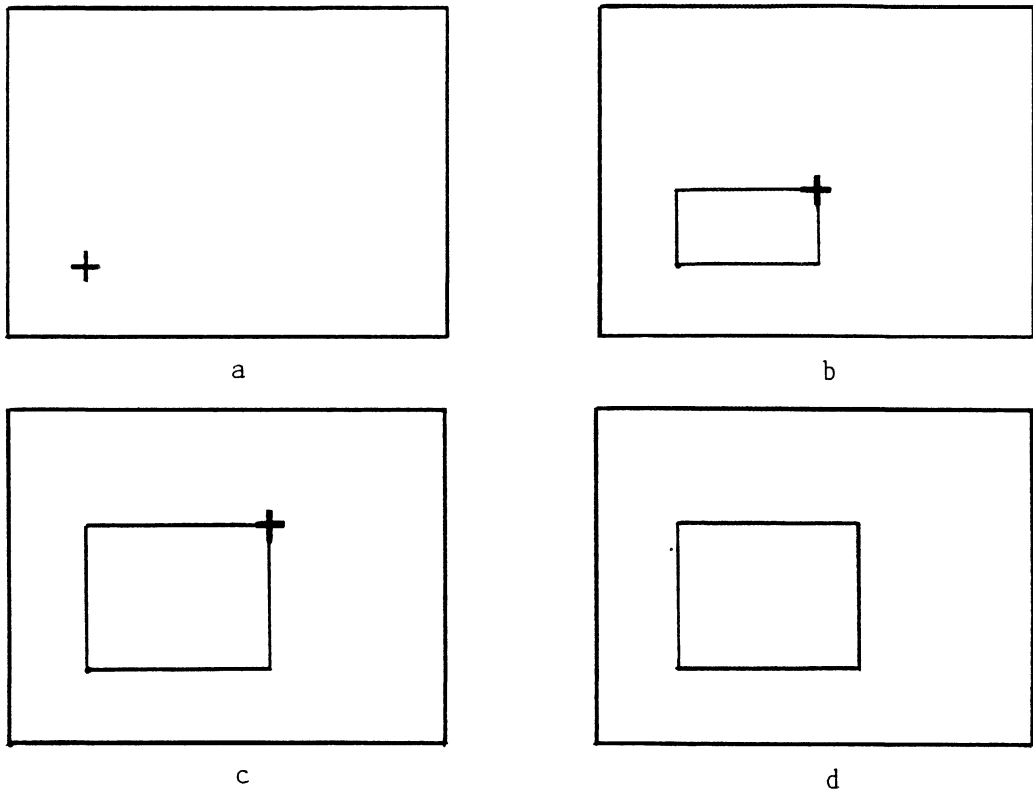
5.2.6 Slepen (dragging).

Bijvoorbeeld bij het tekenen van elektrische schema's wordt wel gebruik gemaakt van een techniek waarbij componenten (een weerstand, een condensator, enz.) in een menu worden aangewezen en vervolgens met de cursor meegenomen worden naar een bepaalde plaats op het scherm. Het lijkt dan alsof de componenten aan de cursor zijn vastgemaakt en door de cursor worden meegesleept. Figuur 5.20 laat deze techniek zien. Het voordeel van deze techniek is, dat het resultaat zichtbaar is voordat een component definitief geplaatst is. Ook voor andere toepassingen kan deze techniek nuttig zijn. Bijvoorbeeld voor het plaatsen van bijschriften in een tekening.

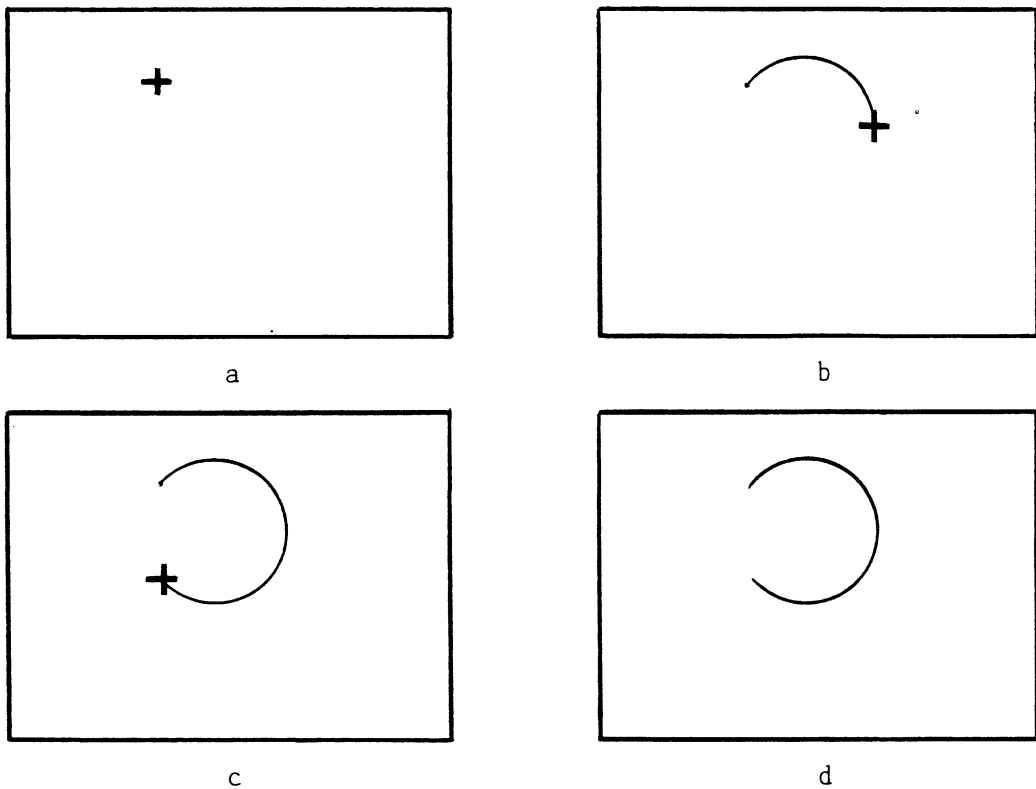
5.3 Invoer toestanden (input modes).

Als in een toepassingsprogramma gegevens worden ingevoerd, kan de interactie tussen het programma en het invoerapparaat op verschillende manieren plaatsvinden:

In REQUEST mode vraagt het toepassingsprogramma een bepaald gegeven in te voeren, en wordt de verdere afwikkeling van het programma gestopt totdat dit gegeven is ingevoerd. Het programma en het invoerapparaat zijn dus om de beurt actief.



Figuur 5.18 Het tekenen van een rechthoek m.b.v. de rubber-band techniek.

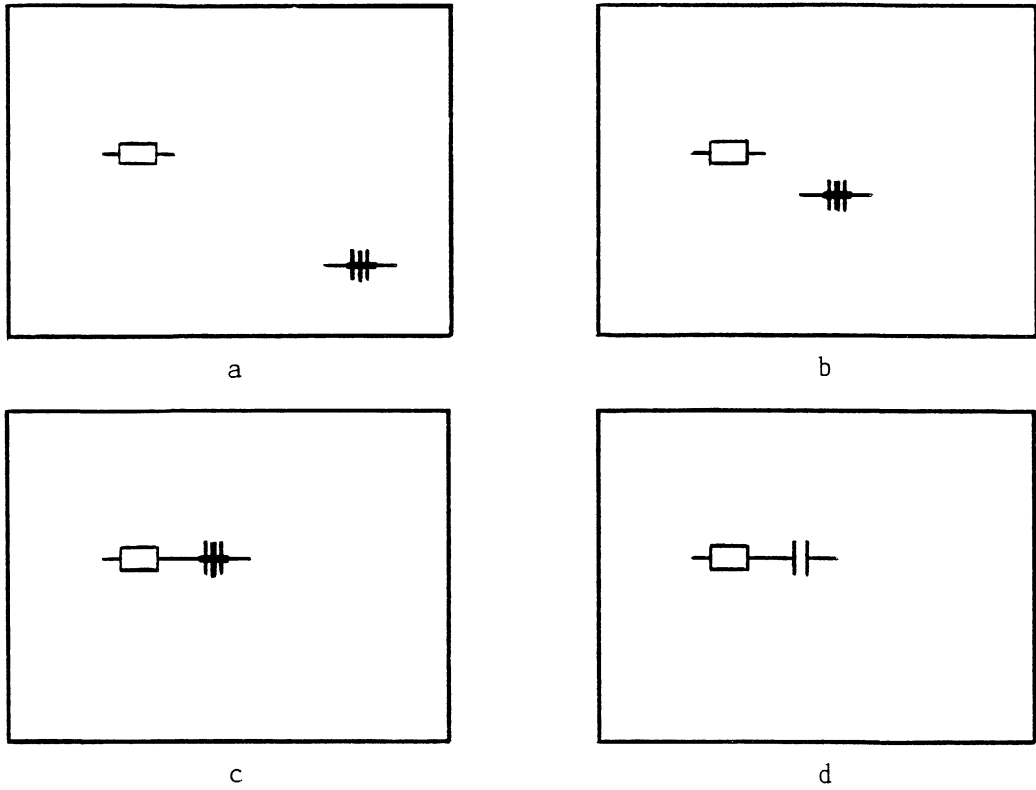


Figuur 5.19 Het tekenen van een cirkelboog m.b.v. de rubber-band techniek.

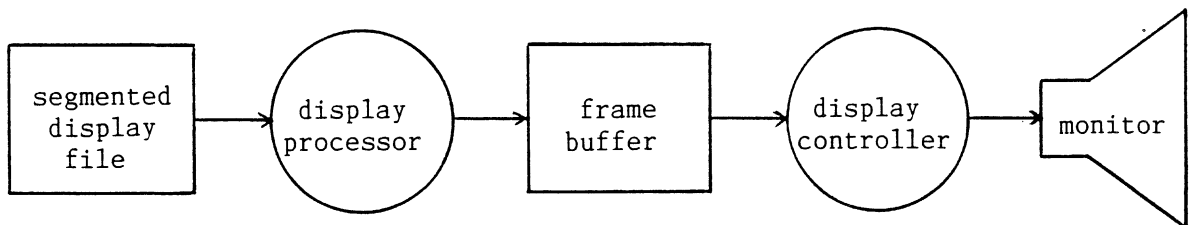
Het invoerapparaat wacht totdat een verzoek wordt gedaan om een gegeven in te voeren. Vervolgens wacht het programma totdat het gegeven is ingevoerd.

In SAMPLE mode zijn het toepassingsprogramma en het invoerapparaat gelijktijdig actief. Het invoerapparaat slaat een bepaald gegeven op en houdt dit gegeven actueel (overschrijft het). Tegelijkertijd gaat het programma verder. Als het programma bepaalde invoergegevens nodig heeft, leest het dat in.

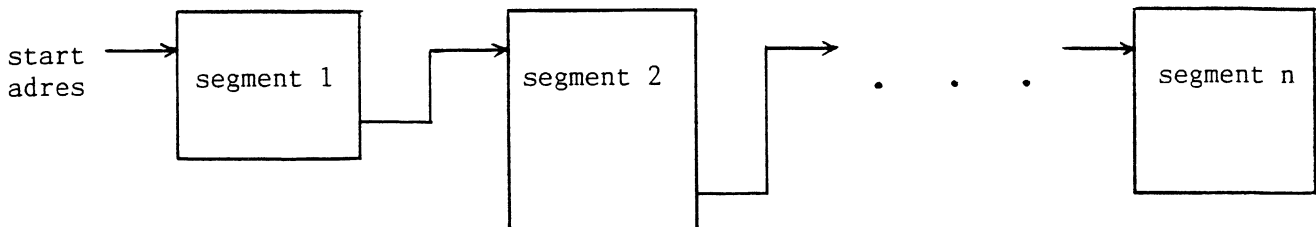
In EVENT mode beginnen één of meerdere invoerapparaten een serie gegevens naar het toepassingsprogramma te sturen. Deze gegevens worden in een wachtrij (input queue) gezet. De gegevens worden één voor één in de volgorde waarin ze zijn ingevoerd door het programma verwerkt. In event mode wordt de afwikkeling van het programma dus gestuurd door het invoerproces.



Figuur 5.20 Het verplaatsen van een element m.b.v. de cursor (dragging).



Figuur 6.1 Displayfile met segmenten bij een rasterbeeldscherm.



Figuur 6.2 Displayfile met segmenten, die m.b.v. een lijststructuur zijn opgeslagen.

6. Segmenten.

6.1 Inleiding.

Een segment bestaat uit een aantal uitvoerprimitieven, die samen een groep vormen, waarop als zelfstandige eenheid operaties uitgevoerd kunnen worden. Stel, bijvoorbeeld, dat een bepaald segment bestaat uit een aantal uitvoerprimitieven die samen een afbeelding van een auto vormen. Dan kan, als men de auto wil verplaatsen, volstaan worden met een verwijzing naar het segment als geheel i.p.v. naar alle uitvoerprimitieven afzonderlijk. Voor veel toepassingen is het efficiënter als een afbeelding wordt opgebouwd m.b.v. segmenten. Bepaalde delen kunnen dan met een enkele opdracht verwijderd of verplaatst worden, terwijl de rest onveranderd blijft.

Wil men dat een aantal uitvoerprimitieven samen een groep vormen, dan moet op de één of andere manier aangegeven worden welke uitvoerprimitieven tot die groep behoren. Een veel gebruikte methode is dat het begin en het einde van een segment worden aangegeven en dat daartussen de uitvoerprimitieven die tot het segment behoren geplaatst worden.

Om naar een segment te kunnen verwijzen, moet het een naam hebben. Vaak is het ook mogelijk om de naam van een segment te veranderen.

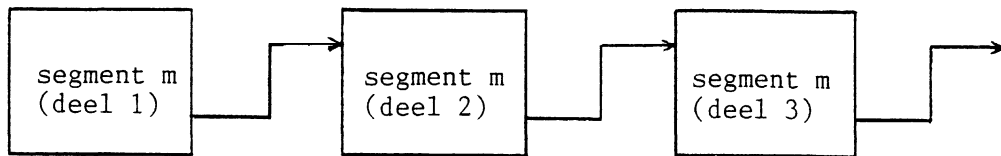
In tekeningen worden vaak bepaalde delen herhaald. Ook zijn bepaalde delen van een tekening vaak symmetrisch. Het is dan efficiënt als een segment (d.w.z. alle uitvoerprimitieven binnen dat segment) gekopieerd en getransformeerd kunnen worden.

6.2 Displayfiles met segmenten.

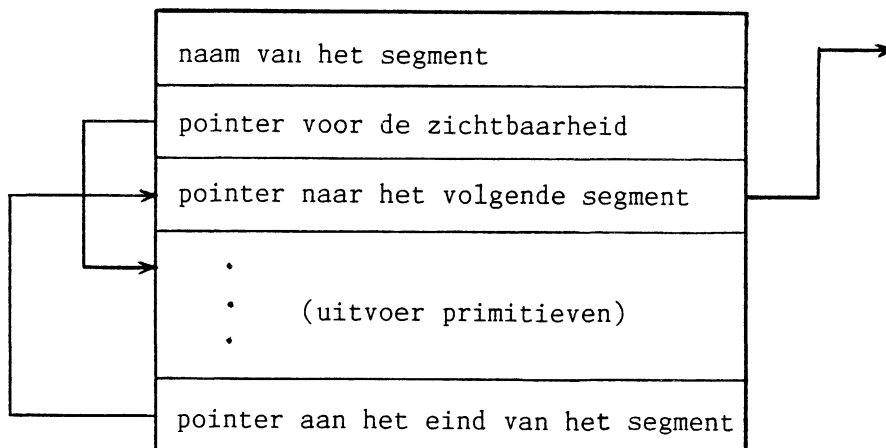
Om het opdelen in segmenten mogelijk te maken, moet de displayfile van een vectorbeeldscherm aangepast worden. De segmented display-file maakt het mogelijk om met segmenten te werken.

Bij een rasterbeeldscherm bevat de framebuffer een representatie van het plaatje op het beeldscherm. De framebuffer bevat geen gegevens over hoe het plaatje wordt opgebouwd. Als een segment dat een ander segment (gedeeltelijk) overlapt, verwijderd wordt, moet het plaatje (gedeeltelijk) opnieuw worden opgebouwd. Daarom wordt ook bij een rastersysteem wel gewerkt met een segmented displayfile (figuur 6.1). Deze displayfile bevat de segmentdefinities en andere informatie betreffende de opbouw van een plaatje. M.b.v. deze file wordt d.m.v. scan-conversie routines de framebuffer gevuld.

Segmented displayfiles kunnen op verschillende manieren opgeslagen worden. Een lijststructuur, zoals in figuur 6.2, wordt veel gebruikt vanwege zijn flexibiliteit. Segmenten kunnen in de gewenste volgorde worden opgenomen en weer verwijderd worden door de pointers aan te passen. Als de



Figuur 6.3 Segment dat over drie geheugenblokken, die een bepaalde vaste grootte hebben, is verdeeld.



Figuur 6.4 Opbouw van een segment.

segmenten t.o.v. elkaar een verschillende prioriteit hebben, kan de volgorde eenvoudig worden aangepast.

Ook voor het toewijzen van de geheugenruimte worden verschillende methoden gebruikt. Door te werken met geheugenblokken die een bepaalde vaste grootte hebben, wordt het beheer van de beschikbare ruimte aanzienlijk vereenvoudigd. Grote segmenten kunnen dan wel over meerdere blokken verdeeld zijn (figuur 6.3).

Figuur 6.4. laat zien hoe een segment kan zijn opgebouwd. Na de naam volgt een veld dat de zichtbaarheid aangeeft. Als een segment niet zichtbaar is, wordt naar het volgende segment verwezen. Is een segment wel zichtbaar, dan volgen de uitvoerprimitieven. In het laatste veld staat een verwijzing naar het derde veld. Hier had ook een verwijzing naar het volgende segment kunnen staan. In dat geval moet echter bij het tussenvoegen en verwijderen van een segment, zowel in het derde als in het laatste veld de pointer worden aangepast. De verwijzing naar het volgende segment is aan het begin geplaatst, waardoor via de ketting snel naar een bepaald segment gezocht kan worden.

Bij een vectorbeeldscherm wordt als een segment gewijzigd moet worden wel gebruik gemaakt van de z.g. dubbele buffering. Een nieuw segment wordt eerst opgeslagen in het geheugen. Daarna wordt het oude segment door het nieuwe vervangen. De geheugenruimte mag pas aan het eind van de refresh cycle worden vrijgegeven, omdat de displayprocessor het oude segment nog aan het lezen kan zijn.

6.3 Segment attributen.

Een segment kan opgevat worden als een zelfstandige eenheid, waaraan een aantal parameters of attributen kunnen worden toegekend die bepalen hoe een segment moet worden afgebeeld. Deze parameters of attributen hebben dan betrekking op alle uitvoerprimitieven die tot het segment behoren.

Een segment kan tijdelijk onzichtbaar worden gemaakt. Als een segment onzichtbaar wordt gemaakt, blijft het opgenomen in de segmented displayfile, maar wordt het niet afgebeeld op het beeldscherm. De mogelijkheid om segmenten al of niet zichtbaar te maken is nuttig in ontwerp-toepassingen waar de effecten van verschillende segmenten beoordeeld moeten worden. Soms is het handig om een menu of een boodschap in een segment op te nemen. Het menu en de boodschap kunnen dan op elk gewenst moment weer zichtbaar worden gemaakt, zonder dat ze opnieuw gedefiniëerd hoeven te worden.

Bij rastersystemen zal men segmenten een verschillende prioriteit willen kunnen toekennen. De segmenten met de laagste prioriteit worden dan het eerst afgebeeld (painter's algoritme). Segmenten met een hogere prioriteit bedekken de segmenten met een lagere prioriteit, als ze elkaar overlappen. Voor animatietoepassingen waar voorwerpen langs elkaar heen bewegen, kunnen zo op een eenvoudige wijze de voorwerpen op de achtergrond en op de voorgrond worden ingesteld.

Soms zal men een bepaald segment willen accentueren (highlighting). De manier waarop een segment geaccentueerd wordt, is afhankelijk van het uitvoerapparaat. Een segment kan bijvoorbeeld met een hogere intensiteit of in een andere kleur worden afgebeeld. Bij een vectorbeeldscherm kan een segment gaan knippen.

Een segment kan ook in zijn geheel getransformeerd worden. Als bij het definiëren van een segment een eenheidsmatrix aan het segment wordt toegekend, kan een segment getransformeerd worden door deze matrix te wijzigen.

Als een plaatje uit veel segmenten is opgebouwd, zal men segmenten willen kunnen herkennen door ze op het beeldscherm aan te wijzen (detecteren).

7. Grafische basisprogrammatuur.

7.1 Inleiding.

Men kan bij grafische programmatuur een aantal niveau's onderscheiden, n.l. systeemprogrammatuur, basisprogrammatuur, toepassingsprogramma's en turnkey-systemen.

Tot de systeemprogrammatuur behoren de besturingsprogramma's voor grafische beeldschermen, printers en plotters (device drivers). Door deze programma's wordt bijvoorbeeld de code voor de besturing van de grafische uitvoerapparatuur aan het interface doorgegeven.

De basisprogrammatuur biedt functies voor het afbeelden en manipuleren van grafische uitvoer en voor afhandeling van grafische invoer. D.m.v. opdrachten in een programma kunnen de functies worden aangeroepen. Basisprogrammatuur is niet gericht op een bepaald soort toepassing, maar dient als hulpmiddel voor het schrijven van toepassingsprogramma's.

De toepassingsprogramma's worden zo ontworpen dat men plaatjes kan afbeelden, zonder zich zorgen te maken over hoe het wordt gedaan. Een gebruiker kan met een dergelijk programma communiceren op een wijze die is aangepast aan een bepaalde toepassing. Voorbeelden van zulke programma's zijn pakketten voor zakelijke toepassingen met faciliteiten voor het maken van grafieken, en pakketten voor ontwerpen en tekenen (CAD).

Een turnkey-systeem omvat een aantal samenhangende toepassingsprogramma's die gericht zijn op de oplossing van een omvangrijker probleem. Meestal wordt de programmatuur compleet met (speciale) apparatuur in een volledig bedrijfsklaar systeem geleverd. Als voorbeeld kan genoemd worden een CAD-systeem gericht op de werktuigbouw, met toepassingsprogramma's voor het invoeren van de vorm van een product, het maken van technische tekeningen en het uitvoeren van sterkteberekeningen.

7.2 Basisprogrammatuur (GKS).

Binnen een pakket dat ontworpen is voor het programmeren van grafische toepassingsprogramma's, zullen functies moeten zijn voor:

- Het afbeelden van de uitvoerprimitieven, zoals punten, lijnen en tekst.
- Het instellen van de intensiteit en de kleur.
- Het toepassen van transformaties.
- Het onderverdelen van plaatjes in segmenten en het uitvoeren van operaties op die segmenten.
- Het afhandelen van de interactie met de invoerapparaten.
- Het opslaan van plaatjes.
- Het afhandelen van fouten.

Daarnaast zullen een aantal huishoudelijke taken verricht moeten kunnen worden, zoals het schoonmaken van een beeldscherm en het initialiseren van de parameters.

Als bij het schrijven van de toepassingsprogrammatuur gebruik wordt gemaakt van gestandaardiseerde grafische functies, kan deze software op verschillende hardware systemen gebruikt worden zonder herschreven te moeten worden. Wel moet een driver aanwezig zijn die de voor een bepaald apparaat geschikte code genereert. Ook zullen niet alle functies met elk apparaat uitgevoerd kunnen worden.

Internationaal hebben allerlei organisaties samengewerkt om een algemeen geaccepteerde standaard voor grafische basisprogrammatuur te ontwikkelen. Na een aanzienlijke krachtsinspanning heeft dit geleid tot de ontwikkeling van het Graphical Kernel System (GKS), dat door de International Standards Organisation (ISO) geaccepteerd is als een grafische software standaard. Alhoewel GKS oorspronkelijk ontworpen is als een twee-dimensionaal grafisch pakket, wordt er inmiddels ook gewerkt aan drie-dimensionale uitbreidingen.

De gestandaardiseerde grafische functies zijn gedefinieerd onafhankelijk van een programmeertaal. Om GKS in een bepaalde programmeertaal te implementeren, moet een taalbinding (language binding) gedefinieerd worden. Deze binding bepaalt o.a. de syntax om de verschillende functies aan te kunnen roepen. Zo specificeert GKS een functie om een aantal met elkaar verbonden lijnstukken af te beelden:

POLYLINE

In PASCAL kan deze functie geïmplementeerd worden als een procedure met de naam `polyline`. Als een toepassingsprogramma in PASCAL wordt geschreven kan de functie worden aangeroepen d.m.v. de opdracht.

```
polyline(n,x,y)
```

Hierin is `n` het aantal punten die met elkaar verbonden worden, en zijn `x` en `y` array's die de coördinaten van deze punten bevatten. Er zijn taalbindingen van GKS gedefinieerd voor o.a. BASIC, FORTRAN, PASCAL en C. Elke taalbinding is zo gedefinieerd, dat er optimaal gebruik wordt gemaakt van de mogelijkheden die de taal biedt om verschillende datatypen vast te leggen, in- en uitvoer- parameters mee te geven en fouten af te handelen.

Alhoewel GKS een specificatie geeft voor grafische basisfuncties, voorziet het niet in een standaard voor een grafische interface voor in- en uitvoerapparaten. Ook worden er geen methoden voor real-time modellering aangegeven. Er zijn voor deze gebieden afzonderlijke standaards ontwikkeld. Het Computer Graphics Interface (CGI) systeem omvat een voorstel voor de standaardisatie van de grafische interface voor uitvoerapparatuur. Een standaardmethode voor real-time modellering wordt gedefinieerd door de Programmer's

Hierarchical Interactive Graphics Standard (PHIGS). Het Computer Graphics Metafile (CGM) systeem specificceert een standaard voor de opslag van beelden.

7.3 Functies binnen GKS.

In deze paragraaf zal een overzicht worden gegeven van een aantal functies zoals die binnen GKS gedefinieerd zijn. De lijst is zeker niet volledig. De bedoeling is slechts om een indruk te geven van de soort functies in grafische basisprogrammatuur. De beschrijvende namen voor de functies zijn gebaseerd op de GKS definities.

7.3.1 Control functies.

Tot de control functies behoren de huishoudelijke taken, zoals het leggen van de juiste verbindingen met de verschillende werkstations, het schoonmaken van het beeldscherm en het initialiseren van de interne parameters:

OPEN GKS

Start het werken met GKS.

CLOSE GKS

Stopt het werken met GKS (de door GKS geclaimde geheugenruimte en de randapparaten worden vrijgegeven).

OPEN WORKSTATION.

Legt een verbinding tussen een bepaald werkstation en GKS.

CLOSE WORKSTATION.

Verbreekt de verbinding tussen dit werkstation en GKS.

ACTIVATE WORKSTATION.

De uitvoer wordt vanaf nu naar dit werkstation gestuurd.

DEACTIVATE WORKSTATION.

De uitvoer wordt niet langer naar dit werkstation gestuurd.

CLEAR WORKSTATION.

Maakt het afbeeldingsgeheugen van een bepaald werkstation schoon (alle binnen het werkstation opgeslagen segmenten worden verwijderd).

REDRAW ALL SEGMENTS ON WORKSTATION.

Maakt het scherm schoon en tekent alle zichtbare segmenten.

7.3.2 Uitvoer functies.

Met de uitvoerfuncties kan een punt, een lijn, een polygoon, een tekst of een marker worden afgebeeld. Een marker is een eenvoudig symbool, zoals een kruisje, een plusteken, een sterretje, enz.

POLYLINE

Tekent een aantal met elkaar verbonden lijnstukken, waarbij de definitiepunten gegeven zijn in wereldcoördinaten (WC). Als er één definitiepunt is, wordt een enkel punt getekend.

POLYMARKER

Tekent een aantal markers van een bepaalde soort op aangegeven plaatsen (in WC).

TEXT

Beeldt een bepaalde tekst af op een aangegeven plaats (in WC). Als een tekst horizontaal wordt afgebeeld, kan de aangegeven plaats bijvoorbeeld de linkeronderhoek van het eerste teken zijn.

FILL AREA

Tekent een polygoon, waarbij de hoekpunten gegeven zijn (in WC). Het polygoon kan ongevuuld blijven, zodat alleen de contour zichtbaar is, of gevuld worden met een kleur, een arcering of een patroon.

GENERALIZED DRAWING PRIMITIVE

Creëert een geometrische entiteit, zoals bijvoorbeeld een cirkel, die dan later meerdere malen kan worden aangeroepen. De figuur wordt d.m.v. een aantal opeenvolgende definitiepunten (in WC) opgeslagen.

7.3.3 Uitvoerattributen.

Attributen zijn de eigenschappen van de uitvoerprimitieven. Zij omvatten de intensiteits- en kleurspecificaties, de lijnsoort, het lettertype, de grootte van de letter en de patronen voor het vullen van gebieden. Als de waarde van een attribuut is ingesteld, dan blijft deze geldig voor alle volgende uitvoerprimitieven, zolang deze waarde tenminste niet veranderd wordt (current value). Niet elke attribuutwaarde kan door elk uitvoerapparaat worden weergegeven (b.v. kleur!). De mogelijke attribuutwaarden kunnen voor een bepaald uitvoerapparaat ingesteld worden. Als er gebruik wordt gemaakt van meerdere uitvoerapparaten, is het handig als de mogelijke attribuutwaarden per werkstation kunnen worden ingesteld. In GKS wordt daarom onderscheid gemaakt tussen werkstation onafhankelijke en werkstation afhankelijke attributen.

7.3.3.1 Werkstation onafhankelijke attributen.

SET LINETYPE

Stelt de lijnsoort (doorgetrokken lijn, streepjeslijn, stippellijn, hartlijn, enz.) in.

SET LINEWIDTH SCALE FACTOR

Stelt de breedte van de lijn in.

SET POLYLINE COLOUR INDEX

Stelt de kleur van een lijn in.

SET MARKER TYPE

Stelt het markertype (x,+,*,æ,∅, enz.) in.

SET MARKER SIZE SCALE FACTOR

Stelt de grootte van een marker in.

SET POLYMARKER COLOUR INDEX

Stelt de kleur van een marker in.

SET TEXT FONT AND PRECISION

Stelt een bepaald lettertype in.

SET CHARACTER EXPANSION FACTOR

Stelt de grootte van een letter in.

SET CHARACTER SPACING

Stelt de afstand tussen de letters in.

SET TEXT COLOUR INDEX
Stelt de kleur van de tekst in.
SET TEXT PATH
Stelt de schrijfrichting van een tekst (van links naar rechts, van boven naar beneden, enz.) in.
SET FILL AREA INTERIOR STYLE
Bepaalt hoe een polygoon gevuld (niet gevuld, egaal gekleurd, gearceerd, patroon) moet worden.
SET FILL AREA STYLE INDEX
Maakt het mogelijk om een bepaald vulpatroon te selecteren.
SET FILL AREA COLOUR INDEX
Bepaalt de kleur van de vulling van een polygoon.

7.3.4 Transformatie-functies.

Een plaatje, dat gedefinieerd is in wereldcoördinaten, wordt eerst getransformeerd naar genormaliseerde apparaatcoördinaten (NDC), voordat de transformatie naar de uiteindelijke apparaatcoördinaten (DC) of schermcoördinaten plaatsvindt. Dit maakt het systeem voldoende flexibel om verschillende uitvoerapparaten te bedienen. Als een tekening geheel binnen het window valt is het clippen overbodig. Door de clipping-routine uit te schakelen kan soms een aanzienlijke tijdwinst worden behaald.

7.3.4.1 Transformatie naar genormaliseerde toestelcoördinaten.

SET WINDOW
Stelt het window (in WC) in.
SET VIEWPORT
Stelt het viewport (in NDC) in.
SELECT NORMALIZATION TRANSFORMATION
Kiest een bepaalde transformatie naar genormaliseerde apparaatcoördinaten.
SET CLIPPING INDICATOR
Stelt het al of niet clippen in.

7.3.4.2 Transformatie naar apparaatcoördinaten.

SET WORKSTATION WINDOW
Stelt het window (in NDC) in.
SET WORKSTATION VIEWPORT
Stelt het viewport (in DC) in.

7.3.5 Segment functies.

7.3.5.1 Functies die het werken met segmenten mogelijk maken.

Plaatjes kunnen onderverdeeld worden in segmenten. Tot de functies die het werken met segmenten mogelijk maken, behoren routines voor het creëren, het veranderen van de naam, en het verwijderen van segmenten. Segmenten kunnen opgeslagen worden in een werkstation-onafhankelijk formaat, de z.g. Workstation Independent Segment Storage (WISS). Een segment dat in de WISS is opgeslagen, kan naar een

werkstation gekopieëerd worden.

CREATE SEGMENT

Opent een bepaald segment, waarna primitieven binnen het segment opgenomen kunnen worden.

CLOSE SEGMENT

Sluit het geopende segment.

RENAME SEGMENT

Verandert de naam van een bepaald segment.

DELETE SEGMENT

Verwijdert een bepaald segment.

COPY SEGMENT TO WORKSTATION

Kopieëert de uitvoerprimitieven binnen een bepaald segment, dat aanwezig is in de WISS, naar een bepaald werkstation.

INSERT SEGMENT

Neemt de uitvoerprimitieven binnen een bepaald segment, dat aanwezig is in de WISS, na eventuele transformaties op in een geopend segment of in de rij van uitvoerprimitieven die niet in een segment zijn opgenomen.

7.3.5.2 Segment attributen.

Segmenten kunnen getransformeerd worden. Als bij het creëren een eenheidsmatrix aan het segment wordt toegekend, kan een segment getransformeerd worden door deze matrix te wijzigen. Segmenten kunnen al of niet zichtbaar worden gemaakt en eventueel geaccentueerd worden. Bovendien kan de prioriteit worden ingesteld. Segmenten kunnen ook detecteerbaar worden gemaakt, zodat ze herkend kunnen worden als ze aangewezen worden met een logisch invoerapparaat van het type PICK.

SET SEGMENT TRANSFORMATION

Stelt een bepaalde transformatiematrix voor een bepaald segment in (de bestaande matrix wordt vervangen door een nieuwe).

SET VISIBILITY

Stelt de zichtbaarheid van een segment in.

SET HIGHLIGHTING

Maakt het mogelijk om een segment te accentueren (b.v. hogere intensiteit, knipperen, enz.).

SET SEGMENT PRIORITY

Stelt de prioriteit van een segment in.

SET DETECTABILITY

Maakt het mogelijk om een segment te detecteren.

7.3.6 Invoer functies.

Bij interactieve grafische toepassingen worden verschillende soorten gegevens (LOCATOR, STROKE, STRING, VALUATOR, PICK, CHOICE) (zie 5.1) op verschillende werkstations met verschillende apparaten (toetsenbord, lichtpen, joystick, muis, tablet, enz.) ingevoerd. Bovendien kan de invoer van gegevens op verschillende manieren plaatsvinden (REQUEST, SAMPLE, EVENT) (zie 5.3).

7.3.6.1 Initialiseren van de invoerapparaten.

Op een bepaald werkstation kan een bepaald invoerapparaat voor het invoeren van bepaalde soorten gegevens geïntialiseerd worden. Bij het initialiseren kan bijvoorbeeld worden vastgelegd wat op het scherm i.v.m. de terugkoppeling wordt afgebeeld (kruisdraad, kruisje, enz.), wat de beginpositie van de cursor is, en wat het gebied is waarbinnen de cursor verplaatst kan worden.

INITIALIZE LOCATOR

Initialiseert op een bepaald werkstation een bepaald apparaat voor het invoeren van de coördinaten van een punt.

Op soortgelijke wijze kan op een bepaald werkstation een bepaald apparaat voor het invoeren van andere soorten gegevens geïntialiseerd worden.

7.3.6.2 Het instellen van de manier waarop de gegevens worden ingevoerd.

SET VALUATOR MODE

Stelt de manier in waarop een getal op een bepaald werkstation m.b.v. een bepaald apparaat zal worden ingevoerd (nl.request, sample of event).

Op soortgelijke wijze kan de manier waarop andere gegevens op een bepaald werkstation m.b.v. een bepaald apparaat worden ingevoerd, ingesteld worden.

7.3.6.3 Invoer en request mode.

REQUEST STROKE

Het toepassingsprogramma vraagt om de coördinaten van een serie punten op een bepaald werkstation m.b.v. een bepaald apparaat in te voeren, en wacht tot dit is gebeurd.

7.3.6.4 Invoer in sample mode.

Op een bepaald werkstation houdt een bepaald invoerapparaat bepaalde gegevens actueel.

SAMPLE STRING

Het toepassingsprogramma vraagt om een tekst, die op een bepaald werkstation m.b.v. een bepaald apparaat is ingevoerd.

Op soortgelijke wijze kan het toepassingsprogramma om andere gegevens vragen die op een bepaald werkstation m.b.v. een bepaald apparaat zijn ingevoerd.

7.3.6.5 Invoer in event mode.

Op verschillende werkstations kunnen m.b.v. verschillende apparaten verschillende gegevens ingevoerd worden. Deze gegevens worden alle in dezelfde invoerrij (input queue)

opgeslagen. De invoerrij kan dus verschillende gegevens bevatten, in de volgorde waarin ze zijn ingevoerd. De gegevens kunnen herkend worden aan de hand van de soort, het werkstation en het invoerapparaat.

AWAIT EVENT

Het toepassingsprogramma kijkt of er gegevens in de invoerrij zijn geplaatst. Als deze gegevens niet aanwezig zijn, wacht het toepassingsprogramma gedurende een bepaalde tijd. Zijn de gegevens daarna nog niet aanwezig, dan gaat het toepassingsprogramma verder zonder deze gegevens. Zijn de betreffende gegevens wel aanwezig, dan worden de gegevens die het langst in de invoerrij hebben gezeten naar het z.g. current event report overgebracht.

GET LOCATOR

Hiermee worden de coördinaten van een punt die in het current event report zitten, aan het toepassingsprogramma doorgegeven.

FLUSH DEVICE EVENTS

Hiermee kunnen bepaalde invoergegevens uit de invoerrij verwijderd worden.

Op soortgelijke wijze kunnen andere gegevens die in het current event report zitten, aan het toepassingsprogramma worden doorgegeven.

7.3.7 Metafile functies.

De gegevens die betrekking hebben op het opbouwen en het afbeelden van plaatjes kunnen voor langere tijd worden opgeslagen in een z.g. metafile.

7.3.8 Inquiry functies.

Als de waarden van variabelen regelmatig veranderen binnen GKS, kan het handig zijn als deze waarden opgevraagd kunnen worden. Binnen GKS zijn verschillende functies aanwezig om de waarden van variabelen op te vragen. Zo kunnen bijvoorbeeld de current values van attributen worden opgevraagd. Ook kan bij een werkstation met een rasterbeeldscherm de grootte en de kleur van de pixels worden opgevraagd.

INQUIRE LINETYPE

Het toepassingsprogramma vraagt naar de lijnsoort zoals die op dat moment op een werkstation is ingesteld.

Op soortgelijke wijze kan het toepassingsprogramma naar andere attribuutwaarden, zoals die op dat moment op een bepaald werkstation zijn ingesteld, vragen.

7.3.9 Utility functies.

EVALUATE TRANSFORMATION MATRIX

Berekent een samengestelde transformatiematrix, die door een fixed point voor de schaling, de schaalfactoren in de x- en de y-richting, een pivot point voor de rotatie, een

rotatiehoek en een translatievector bepaald wordt, en voert de resulterende transformatiematrix uit.

ACCUMULATE TRANSFORMATION MATRIX

Voert een gegeven transformatiematrix in, vermenigvuldigt deze met een samengestelde matrix, die door een fixed point voor de vermenigvuldiging, de vermenigvuldigingsfactoren in de x- en de y-richting, een pivot point voor de rotatie, een rotatiehoek en een translatievector bepaald wordt en voert de resulterende transformatiematrix uit.

7.3.10 Foutafhandeling.

ERROR LOGGING

Hiermee kan een foutmelding met de naam van de betreffende functie worden afgedrukt.

7.4 Programma met GKS-functies.

Als voorbeeld volgt nu een programma in PASCAL, waarbij gebruik wordt gemaakt van GKS-functies. De betekenis van de nummers voor de kleur en het vulpatroon zijn afhankelijk van de mogelijkheden van het uitvoerapparaat. De nummers kunnen daarom voor elk werkstation een andere betekenis hebben.

```
program voorbeeld (input,output);

{$I gks.inc}

type
  pos=record
    x:real;
    y:real
  end;
  ea=record
    xl:real;
    yl:real;
    xr:real;
    yr:real
  end;
var
  x,y:array[1..10] of real;
  start_position:pos;
  echo_area:ea;
  tx,ty:real;
  m:array[1..3,1..3] of real;

begin
  (*openen*)
  open_gks;
  open_workstation(5);
  activate_workstation(5);

  (*instellen window en viewport*)
  set_window(1,0,0,400,300);
  set_viewport(1,0,0,1,1);
  select_normalization_transformation(1);
```



```
set_workstation_window(5,0,0,1,1);
set_workstation_viewport(5,0,0,640,480);

(*opbouwen segment 1*)
create_segment(1);
x[1]:=250; y[1]:=50;
x[2]:=300; y[2]:=150;
set_linetype(dotted);
set_linewidth_scale_factor(2);
set_polyline_colour_index(5);
polyline(2,x,y);
close_segment;

(*opbouwen segment 2*)
create_segment(2);
x[1]:=100; y[1]:=100;
x[2]:=200; y[2]:=100;
x[3]:=200; y[3]:=200;
x[4]:=100; y[4]:=200;
set_fill_area_interior_style(solid);
set_fill_area_interior_colour_index(3);
fill_area(4,x,y);
set_character_expansion_factor(1.5);
set_text_colour_index(4);
set_text_path(down);
text(80,180,"vierkant");
close_segment;

(*invoeren translatievector*)
start_position.x:=200; start_position.y:=150;
echo_area.xl:=0; echo_area.yl:=0;
echo_area.xr:=640; echo_area.yr:=480;
initialize_locator(5,2,start_position,1,echo_area);
(*nummer 2 is tablet en nummer 1 betekent verplaats de kruisdraad
m.b.v. de handcursor en voer de coördinaten in door op de gele
toets te drukken*)
set_locator_mode(5,2,request);
request_locator(5,2,tx,ty);

(*verplaatsen van segment 1*)
evaluate_transformation_matrix(0,0,1,1,0,0,0,tx,ty,m);
set_segment_transformation(1,m);
redraw_all_segments_on_workstation(5);

(*sluiten*)
deactivate_workstation(5);
close_workstation(5);
close_gks
end.
```

