

Informationsverarbeitungssysteme
Graphische Datenverarbeitung
Graphisches Kernsystem für drei Dimensionen (GKS-3D)
Funktionale Beschreibung
Identisch mit ISO 8805 : 1988

DIN
ISO 8805

Information processing systems; Computer graphics; Graphical Kernel System for Three Dimensions (GKS-3D) functional description; Identical with ISO 8805 : 1988

Systèmes de traitement de l'information; Infographie; Système graphique de base pour trois dimensions (GKS-3D) description fonctionnelle; Identique à ISO 8805 : 1988

Die Internationale Norm ISO 8805 : 1988 „Information processing systems — Computer graphics — Graphical Kernel System for Three Dimensions (GKS-3D) functional description“ ist unverändert in diese Deutsche Norm übernommen worden.

Nationales Vorwort

(1) Einführung

Die Internationale Norm wurde im ISO/TC 97/SC 21/WG 2 (jetzt ISO/IEC JTC 1/SC 24*) unter Mitwirkung des Ausschusses „Graphische Datenverarbeitung“ des Normenausschusses Informationsverarbeitungssysteme (NI) erarbeitet.

Das Präsidium des DIN hat mit Präsidialbeschluß 13/1983 festgelegt, daß DIN-Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründet, in Ausnahmefällen und unter bestimmten Bedingungen in Englisch veröffentlicht werden dürfen. Diese Bedingungen sind für diese Norm erfüllt.

Um den Gebrauch dieser Norm zu erleichtern, wurde dem englischen Originaltext eine englisch-deutsche Fachwörterliste vorangestellt.

In dieser Fachwörterliste werden den englischen Benennungen deutsche Benennungen und Erläuterungen zugeordnet. Die Zuordnung gilt nur für diese Norm und für deren Anwendungsbereich. Gleiche fremdsprachige Benennungen können auch in anderen Fachwörterlisten auftreten und dort anderen deutschen Benennungen zugeordnet sein. Deshalb hat diese Fachwörterliste keine allgemeine terminologische Verbindlichkeit.

(2) Einleitung

Das Graphische Kernsystem für drei Dimensionen (GKS-3D) beschreibt eine Menge von Funktionen für

- a) die Definition und Darstellung von zwei- und dreidimensionalen (2D und 3D) graphischen Daten;
- b) die Speicherung und Bearbeitung von graphischen Daten;
- c) die Eingabe von Daten mit graphischem Bezug.

*) ISO: International Organization for Standardization
IEC: International Electrotechnical Commission
TC 97: Information Processing Systems
JTC 1: Joint Technical Committee „Information Technology“
SC 21: Open Systems
SC 24: Computer Graphics
WG 2: Computer Graphics

Die Hauptgründe für die Einführung dieser Internationalen Norm für die graphische Datenverarbeitung sind:

- d) eine leichte Übertragbarkeit von Anwendungsprogrammen mit graphischen Komponenten zwischen verschiedenen Installationen zu erlauben;
- e) das Verständnis und die Benutzung graphischer Methoden durch Anwendungsprogrammierer zu erleichtern;
- f) den Herstellern graphischer Geräte Richtlinien für sinnvolle Kombinationen von graphischen Fähigkeiten in einem Gerät zu geben.

Um diese Hauptziele zu erreichen, berücksichtigt die Konzeption des GKS-3D folgende Anforderungen:

- g) GKS-3D soll alle Fähigkeiten beinhalten, die für das ganze Spektrum der graphischen Datenverarbeitung — von einfacher, passiver Ausgabe bis hin zu hochgradig interaktiven Anwendungen — erforderlich sind.
- h) Der ganze Bereich graphischer Geräte, Vektor- und Rastergeräte, Mikrofilmaufnahmegeräte, Speicherbildschirme, Bildschirme mit Bildauffrischung und Farbsichtgeräte, muß vom GKS-3D auf einheitliche Art und Weise bedient werden.
- i) GKS-3D muß alle Fähigkeiten bereitstellen, die bei der Mehrzahl der Anwendungen erforderlich sind.

Diese Anforderungen wurden zur Formulierung einer Anzahl von Prinzipien verwendet, die zur Beurteilung der wesentlichen Alternativen herangezogen wurden. Dadurch war es möglich, bei der Entscheidung von Detailfragen die globale Zielsetzung im Auge zu behalten. Fünf Gesichtspunkte wurden herausgearbeitet, die jeweils eine Gruppe von Prinzipien zusammenfassen:

- j) Entwicklungsziele: Die folgenden Prinzipien durften durch technische Entwicklungsentscheidungen nicht verletzt werden:
 - 1) Konsistenz: Keine der Pflichtenforderungen des GKS-3D dürfen sich gegenseitig widersprechen.
 - 2) Verträglichkeit: Keine anderen Normen oder allgemein akzeptierte Regeln der Praxis dürfen verletzt werden.
 - 3) Orthogonalität: Die Funktionen oder Moduln des GKS-3D sollten unabhängig voneinander sein oder die Abhängigkeit muß strukturiert und genau bezeichnet sein.

Fortsetzung Seite 2 bis 10
und 386 Seiten ISO-Norm

Normenausschuß Informationsverarbeitungssysteme (NI) im DIN Deutsches Institut für Normung e.V.

Jede Art der Vervielfältigung, auch auszugsweise, nur mit Genehmigung des DIN Deutsches Institut für Normung e.V., Berlin, gestattet.

k) Funktionale Fähigkeiten: Die folgenden Prinzipien legen den Leistungsumfang des GKS-3D fest:

- 1) Vollständigkeit: Alle Funktionen müssen vorhanden sein, die die Mehrzahl der Anwendungen bei einer gegebenen Leistungsstufe der Funktionalität braucht.
- 2) Minimalität: Es sollen keine Funktionen angegeben werden, die für die Anwendungen einer gegebenen Leistungsstufe der Funktionalität unnötig sind.
- 3) Knappheit: Eine Anwendung muß imstande sein, ein gewünschtes Ergebnis durch eine Menge von Funktionen und Parametern zu erreichen, die so klein wie möglich ist.
- 4) Reichhaltigkeit: Eine reichhaltige Menge von Funktionen bietet umfassende Möglichkeiten, die über die Grundfunktionen hinausgehen und höhere Fähigkeiten beinhalten.

Es ist klar, daß zwischen den Prinzipien in dieser Gruppe Kompromisse eingegangen werden müssen. Daher werden die GKS-3D-Funktionen in neun Leistungsstufen zusammengefaßt. Eine GKS-3D-Implementierung enthält genau die Funktionen einer dieser Leistungsstufen. Während die niedrigste Leistungsstufe nur eine Minimalmenge von Funktionen enthält, erlauben die höheren Leistungsstufen eine Erweiterung über die Grundbedürfnisse hinaus in Richtung größerer Reichhaltigkeit.

l) Benutzerschnittstelle: Die folgenden Prinzipien wurden bei der Definition der Benutzerschnittstelle herangezogen:

- 1) Benutzerfreundlichkeit: GKS-3D soll den Entwurf einer wünschenswerten Benutzerschnittstelle erlauben.
- 2) Klarheit: Die Konzeption und die funktionalen Fähigkeiten des GKS-3D sollen, besonders für den Anwendungsprogrammierer, leicht verständlich sein.
- 3) Fehlerbehandlung: Versagen von Systemfunktionen oder Modulen, die durch Fehler des Systems selber oder vom Anwendungsprogramm verursacht werden, sollen so behandelt werden, daß die Fehlerreaktion leicht verständlich und für den Anwendungsprogrammierer informativ ist und daß der Einfluß auf das System und das Anwendungsprogramm so klein wie möglich ist.

Klarheit und eine gute Fehlerbehandlung sind wichtig für die Benutzerfreundlichkeit. Die Spezifikation der Fehlerbehandlung ist ein integraler Bestandteil des GKS-3D. Das System und sein Zustand können zur Unterstützung der Klarheit dem Benutzer auf leicht verständliche Weise dargestellt werden.

Klarheit wurde nicht nur auf den Systementwurf, sondern auch auf die Systembeschreibung angewandt. Zu diesem Zwecke wurde die GKS-3D-Spezifikation in eine allgemeine Beschreibung, eine Beschreibung der zugrundegelegten, logischen Datenstrukturen zur Darstellung der Zustände des Systems und eine Beschreibung der Funktionen und ihre Wirkung auf diese Datenstrukturen aufgeteilt.

m) Graphische Geräte: Die folgenden Prinzipien sind der Menge der graphischen Geräte, die vom GKS-3D bedient werden können, zugeordnet:

- 1) Geräteunabhängigkeit: GKS-3D-Funktionen sollen entworfen sein, um dem Anwendungsprogramm bei Benutzung dieser Funktionen die Verwendung der Fähigkeiten von ganz verschiedenen graphischen Ausgabe- und Eingabegeräten ohne Änderung der Programmstruktur zu erlauben.

- 2) Gerätevielfalt: Alle Fähigkeiten einer großen Vielfalt der verschiedenen graphischen Ausgabe- und Eingabegeräte sollen durch die GKS-3D-Funktionen zugänglich sein.

Diese Prinzipien haben zur fundamentalen Konzeption der GKS-3D-Architektur geführt: Die Konzeption mehrerer unabhängiger graphischer Arbeitsplätze, die mit dem GKS-3D verbunden und durch das GKS-3D betrieben werden. Das Anwendungsprogramm kann die Fähigkeiten jedes graphischen Arbeitsplatzes erfragen. GKS-3D enthält Fluchtfunktionen, die im Anwendungsprogramm leicht zu erkennen sind, und die zur Verwendung besonderer Fähigkeiten eines bestimmten Gerätes benutzt werden können.

n) Implementierung: Die letzte Gruppe der Prinzipien steht im Zusammenhang mit der GKS-3D-Implementierung:

- 1) Implementierbarkeit: Es soll möglich sein, die GKS-3D-Funktionen in den meisten Gastsprachen, mit den meisten Betriebssystemen und bei den meisten graphischen Geräten zu unterstützen.
- 2) Sprachunabhängigkeit: Es soll möglich sein, die genormten Fähigkeiten des GKS-3D von allen genormten Programmiersprachen ansprechen zu können.
- 3) Leistungsfähigkeit: Die Norm soll ohne zeitintensive Algorithmen implementiert werden können.
- 4) Robustheit: Der Bediener und der Anwendungsprogrammierer sollen auf bestmögliche Art gegen Versagen von Hardware und Software des Systems geschützt sein.

Die fünf Gruppen von Prinzipien können nicht getrennt voneinander gesehen werden. Beispielsweise tragen Ziele und funktionale Fähigkeiten zur Benutzerfreundlichkeit bei. Leistungsfähigkeit ist außerdem wichtig, wenn man die Antwortzeiten in einer interaktiven Umgebung betrachtet. Einige Prinzipien widersprechen sich, wie Reichhaltigkeit gegen Minimalität, umfassende Fehlerbehandlung gegen Leistungsfähigkeit und Knappheit gegen Gerätevielfalt. Kompromisse müssen geschlossen werden, damit die globalen Ziele erreicht werden. GKS-3D soll eine leicht verständliche Struktur und eine Menge von Funktionen haben, die die große Mehrzahl der Benutzer graphischer Datenverarbeitung in die Lage versetzt, übertragbare und geräteunabhängige Anwendungsprogramme für das gesamte Spektrum der graphischen Geräte zu entwerfen.

o) Beziehungen zu ISO 7942: Diese Internationale Norm beachtet in bezug auf ISO 7942 die folgenden Entwicklungsziele:

- 1) Die Beziehungen zwischen den Funktionen in ISO 7942 und den Funktionen, die in dieser Internationalen Norm beschrieben sind, sollen wohldefiniert und nicht implementierungsabhängig sein.
- 2) Das Format der Funktionen und ihrer Parameter in ISO 7942 wird unverändert in dieser Norm übernommen.
- 3) Zusätzliche Funktionen werden nur für dreidimensionale Funktionalität eingeführt.

p) Kompatibilität von Anwendungsprogrammen: Ein Hauptziel dieser Internationalen Norm ist es, daß Anwendungsprogramme, die Funktionen von ISO 7942 benutzen, ohne Änderungen auf Implementierungen laufen, die dieser Norm entsprechen, und auf gleichen Arbeitsplätzen die gleichen Resultate erzeugen, abgesehen von den zugelassenen Unterschieden gemäß Anhang D. Es wird jedoch eingeräumt,

daß dieses Ziel nicht immer erreicht werden kann, zum Beispiel bei Benutzung von Bilddateien (weitere Hinweise in Anhang D).

- q) Beziehungen zu ISO 9592: Diese Internationale Norm hat zu ISO 9592 (PHIGS) folgende Beziehung:

Das primäre Ziel von PHIGS und GKS/GKS-3D ist verschieden. Im Unterschied zu GKS/GKS-3D ist PHIGS nicht auf Graphik allein beschränkt, sondern stellt Funktionen zur Definition und zum Editieren von Datenstrukturen bereit, die nicht notwendigerweise Bezug zu graphischer Ausgabe haben. PHIGS bietet Möglichkeiten zur Datenstrukturierung an, die zur graphischen Ausgabe benutzt werden können, aber nicht ausschließlich für graphische Ausgabe und Interaktion gedacht sind.

GKS/GKS-3D ist auf die unmittelbare Erzeugung von graphischen Darstellungselementen ausgerichtet, die in Segmenten zusammengefaßt werden können. PHIGS erzeugt eine Datenstruktur, die durchlaufen werden muß, um graphische Ausgabe zu erzeugen. Jedoch sind die Darstellungselemente und ihre Attribute, die während des Durchlaufs erzeugt werden, fast vollständig identisch zu denen in GKS-3D.

PHIGS legt das gleiche Eingabemodell zugrunde wie GKS-3D. Der Maßwert des logischen Eingabegeräts Picker ist erweitert worden, damit das Strukturelement im zentralen Strukturspeicher von PHIGS eindeutig identifiziert werden kann, das das Darstellungselement erzeugt, welches der Bediener gepickt hat.

Die Anhänge C bis H dienen nur zur Information. Sie sind nicht Teil der Norm.

(3) Anwendungsbereich und Zweck

Diese Internationale Norm spezifiziert eine Menge von Funktionen für die Programmierung in der graphischen Datenverarbeitung, das Graphische Kernsystem für drei Dimensionen (GKS-3D). GKS-3D bildet ein graphisches Grundsystem für Anwendungen, die in der Rechenanlage erzeugte dreidimensionale Bilder auf graphischen Ausgabegeräten darstellen. Sie unterstützt Bediener-eingabe und Interaktion durch Grundfunktionen für die graphische Eingabe und Bildsegmentierung. Sie erlaubt die Speicherung und dynamische Veränderung von Bildern. Ein grundlegendes Prinzip im GKS-3D ist die Zusammenfassung von graphischen Geräten zu einem graphischen Arbeitsplatz, der aus einer Anzahl von Eingabegeräten und einem einzigen Ausgabegerät besteht. Mehrere graphische Arbeitsplätze können gleichzeitig bedient werden. Das Anwendungsprogramm kann sein Verhalten an den graphischen Arbeitsplatz anpassen, um dessen Fähigkeiten besonders gut zu nutzen. Diese Internationale Norm enthält auch Funktionen zur Speicherung von Bildern in einer externen Bilddatei und zu ihrer Verarbeitung. Schließlich sind die Funktionen in aufwärtskompatiblen Leistungsstufen organisiert.

Diese Internationale Norm stellt Anwendungsprogrammen Funktionen zur Verfügung, um dreidimensionale (3D) graphische Darstellungselemente zu definieren und darzustellen, die mit dreidimensionalen Koordinaten beschrieben werden. Zusätzlich erlaubt das Eingabemodell von GKS-3D dreidimensionale Eingabe von Positionen und Linien.

Die Darstellungselemente in dieser Internationalen Norm unterscheiden sich von denen in ISO 7942 folgendermaßen:

- a) Die Darstellungselemente Linienzug und Polymarke werden durch eine beliebige Folge von 3D-Koordinaten definiert.

- b) Die Darstellungselemente Text, Füllgebiet und Zellmatrix bleiben eben (planar), können aber auf einer beliebigen Ebene im 3D-Raum liegen.
c) Das Darstellungselement Füllgebietmenge wird neu eingeführt.

In dieser Internationalen Norm werden Funktionen bereitgestellt, um diese Darstellungselemente zu erzeugen.

Die Funktionen aus ISO 7942 können ebenfalls in dieser Norm verwendet werden. Jedoch erzeugen die Ausgabefunktionen 3D-Darstellungselemente, die in einer Ebene liegen. Zusätzlich werden in dieser Internationalen Norm Fähigkeiten bereitgestellt, um:

- d) Ansichten von 3D-Objekten zu spezifizieren,
e) Informationen von 3D-Eingabegeräten zu erhalten,
f) verdeckte Linien und Flächen bei der Darstellung auf graphischen Arbeitsplätzen zu unterdrücken.

Diese Internationale Norm stellt keine besonderen Funktionen bereit, um Darstellungen unter Berücksichtigung von Lichtquellen, Schattierungen, Mustern und Schattwurf zu erzeugen. Jedoch ist es erlaubt, daß graphische Arbeitsplätze lokal diese Techniken beim Erzeugen von Darstellungselementen und Auswertung von Attributen einsetzen. Dabei müssen die allgemeinen Richtlinien für erlaubte Unterschiede auf graphischen Arbeitsplätzen in Anhang D beachtet werden.

Für bestimmte Funktionsparameter definiert GKS-3D einen Wertebereich, der für die Registrierung oder zukünftige Normung (siehe Abschnitt 4.1.2) reserviert ist. Die Bedeutung dieser Werte wird durch Registrierung festgelegt.

GKS-3D definiert den sprachunabhängigen Kern eines graphischen Systems. Zur Integration in eine Programmiersprache muß GKS-3D in eine sprachabhängige Schicht eingebettet werden, die den besonderen Eigenarten dieser Sprache gehorcht.

(4) Begriffe

Dieser Abschnitt enthält die Definition von Begriffen, die in dieser Internationalen Norm verwendet werden.

Anmerkung: Soweit wie möglich wurde die allgemein akzeptierte Terminologie der graphischen Datenverarbeitung verwendet.

4.1 acknowledgment — Quittung

Eine Anzeige für den Bediener eines logischen Eingabegerätes, die das Betätigen des Auslösers anzeigt.

4.2 aspect ratio — Aspektverhältnis

Das Verhältnis Längen auf den Hauptachsen eines Objekts.

4.3 aspect source flag ASF — Aspektanzeiger

Ein Anzeiger, der angibt, ob ein bestimmter geräteabhängiger Aspekt eines Darstellungselements über ein Attributsbündel oder durch individuelle Attribute ausgewählt wird.

4.4 aspects of primitives — Aspekte von Darstellungselementen

Einflußgrößen, die die Variationsbreite im Aussehen eines Darstellungselements bestimmen. Einige Aspekte werden direkt durch Darstellungsattribute, andere indirekt durch eine Bündeltabelle gesteuert. Aspekte von Segmenten (z.B. Hervorheben) wirken sich nur auf in ihnen enthaltene Darstellungselemente aus. Darstellungselemente außerhalb von Segmenten werden durch diese Aspekte nicht beeinflusst.

4.5 attribute — Attribut

Attribute steuern die Eigenschaften von Darstellungselementen und Segmenten. Es gibt vier Arten von Darstellungsattributen: geometrische, nichtgeometrische, Abbildungs- und Identifizierungsattribute. Die geometrischen und nichtgeometrischen Attribute steuern die Werte der Aspekte der Darstellungselemente.

4.6 back plane — Hintere Klippebene

Eine Ebene parallel zur Projektionsebene, dessen Lage als N-Koordinate im Referenzkoordinatensystem der Darstellungstransformation spezifiziert wird. Darstellungselemente hinter dieser Ebene liegen außerhalb des Darstellungskörpers.

4.7 baseline — Schriftlinie

Eine horizontale Linie im Zeichenkörper (siehe Bild 3), die für viele Zeichendefinitionen die untere Begrenzung des Zeichens bedeutet. Eine kleine Anzahl von Zeichen verläuft unterhalb dieser Linie. Alle Schriftlinien einer Schriftart haben dieselbe Position innerhalb eines Zeichenkörpers.

4.8 bundle index — Bündelindex

Ein Index in einer Bündeltabelle für ein bestimmtes Darstellungselement. Er definiert die geräteabhängigen Aspekte des Darstellungselements.

4.9 bundle table — Bündeltabelle

Eine geräteabhängige Tabelle, die einem bestimmten Darstellungselement zugeordnet ist. Die Einträge in dieser Tabelle geben alle geräteabhängigen Aspekte eines Darstellungselements an. Im GKS-3D existieren Bündeltabellen für die Darstellungselemente Linienzug, Polymark, Text und Füllgebiet.

4.10 capline — Versalhöhe

Eine horizontale Linie im Zeichenkörper (siehe Bild 3), die für viele Zeichendefinitionen die obere Begrenzung des Zeichens bedeutet. Eine kleine Anzahl Zeichen verläuft oberhalb dieser Linie. In einigen Sprachen gibt es zusätzliche Markierungen über den Zeichen (z.B. ein Akzent), die oberhalb der Versalhöhe definiert werden können. Alle Versalhöhen einer Schriftart haben dieselbe Position innerhalb eines Zeichenkörpers.

4.11 cell array — Zellmatrix

Ein GKS-3D-Darstellungselement, das aus einem rechteckigen Netz gleich großer Rechteckzellen besteht, von denen jede eine individuelle Farbe haben kann.

4.12 centre of projection — Projektionszentrum

Siehe Projection Reference Point — Projektionsreferenzpunkt.

4.13 centreline — Mittellinie

Eine vertikale Linie, die den Zeichenkörper halbiert (siehe Bild 3).

4.14 character base vector — Zeichenbasisvektor

Ein Textaspekt, der die Hauptrichtung einer Zeichenfolge definiert. Es ist ein zweidimensionaler Vektor in der Textebene, die im Textdarstellungselement definiert wird.

4.15 character body — Zeichenkörper

Ein Rechteck, das der Entwerfer einer Schriftart zum Definieren der Zeichen verwendet (siehe Bild 3). Alle Zeichenkörper einer Schriftart haben dieselbe Höhe.

4.16 character expansion factor — Zeichenbreitfaktor

Ein Textaspekt, der die Abweichung der Zeichenbreite von dem definierten Nominalwert angibt.

4.17 character height — Zeichenhöhe

Ein Textaspekt, der die nominale Höhe eines Großbuchstaben (Versal) angibt.

4.18 character set — Zeichensatz

Eine registrierte Interpretation für Einträge in der Zeichen-Kodiertabelle (siehe ISO 2022).

4.19 character spacing — Zeichenabstand

Ein Textaspekt, der den Bruchteil der nominalen Zeichenhöhe eines Großbuchstaben angibt, der zwischen zwei nebeneinanderliegende Zeichen eingefügt werden soll.

4.20 character up vector — Zeichenaufwärtsvektor

Ein Textaspekt, der die Hauptaufwärtsrichtung einer Zeichenfolge definiert. Es ist ein zweidimensionaler Vektor in der Textebene, die im Textdarstellungselement definiert wird.

4.21 character width — Zeichenbreite

Ein Textaspekt, der die nominale Breite eines Großbuchstaben angibt.

4.22 choice device — Auswähler

Ein logisches Eingabegerät des GKS-3D, das eine nicht-negative ganze Zahl liefert. Die Zahl stellt eine Auswahl aus mehreren Wahlmöglichkeiten dar.

4.23 CIE — CIE

Abkürzung für Commission International de L'Eclairage. Zeigt an, daß das universelle Farbdefinitionssystem der CIE als Farbmodell benutzt wird.

4.24 CIE L*u*v* colour model — CIE L*u*v* Farbmodell

Ein Farbmodell mit einem ungefähr gleichmäßigen Farbraum für kleine Farbdifferenzen und einer entsprechenden Farbdifferenzformel. Seine Komponenten sind L* (Helligkeit) und u*, v* (Farbe). Die Koordinaten des zugehörigen Farbdigramms sind u* und v*.

4.25 CIE Tristimulus colour model — CIE Tristimulus-Farbmodell

Ein Farbmodell, das es erlaubt, alle Farben im Farbbereich durch nichtnegative Tristimulus-Farbwerte X, Y, Z zu beschreiben.

4.26 clipping — Klippen

Das Entfernen derjenigen Teile von Darstellungselementen, die außerhalb einer angegebenen Grenze (üblicherweise eines Fensters oder eines Darstellungsfeldes) liegen.

4.27 clipping volume — Klippkörper

Entweder das Darstellungsfeld der Normierungstransformation, der Klippkörper der Darstellungstransformation oder der Durchschnitt von beiden. Das Darstellungsfeld der Normierungstransformation ist ein rechtwinkliges Parallelepiped im normierten Koordinatensystem (NDC). Der Klippkörper der Darstellungstransformation liegt im Referenzkoordinatensystem der Darstellungstransformation und wird definiert durch Fenster, hintere und vordere Klippebene und Projektionsstrahlen durch die Ecken des Fensters. Darstellungselemente innerhalb dieses Körpers werden nicht geklippt. Teile von Darstellungselementen, die auf den begrenzenden Ebenen dieser Körper liegen, werden als innerhalb liegend betrachtet.

4.28 colour model — Farbmodell

Die Methode, durch die das Anwendungsprogramm Farben beschreibt.

4.29 colour table — Farbtabelle

Eine geräteabhängige Tabelle, in der jeder Eintrag die Intensitäten von Rot, Grün und Blau angibt und damit eine bestimmte Farbe definiert.

4.30 coordinate graphics, line graphics — Koordinatengraphik, Liniengraphik

Graphische Datenverarbeitung, bei der die Bilder aus Darstellungskommandos und Koordinaten erzeugt werden.

4.31 device coordinate (DC) — Gerätekoordinate

Ortsangabe in einem Koordinatensystem, das geräteabhängig ist. Im GKS-3D sind für ein Gerät, das genaues Skalieren des Bildes erlaubt, die Einheiten des Gerätekoordinatensystems Meter, sonst sind sie geräteabhängige Einheiten.

4.32 device driver — Gerätetreiber

Der geräteabhängige Teil einer GKS-3D-Implementierung zur Unterstützung eines graphischen Gerätes. Der Gerätetreiber erzeugt geräteabhängige Ausgabe und verarbeitet geräteabhängige Eingabe.

4.33 device space — Gerätebereich

Der Bereich, der durch die Menge der adressierbaren Punkte eines graphischen Gerätes festgelegt wird.

4.34 display device, graphics device — Bildschirm, graphisches Gerät

Ein Gerät (z.B. Bildschirm mit Bildauffrischung, Speicherbildschirm, Plotter), auf dem Bilder dargestellt werden können.

4.35 display image, picture — Graphische Darstellung, Bild

Die Menge der Darstellungselemente oder Segmente, die zu einem Zeitpunkt auf der Darstellungsfläche sichtbar sind.

4.36 display space — Darstellungsbereich

Der Teil des Gerätebereichs, der dem für das Bild zur Verfügung stehenden Teil der Darstellungsfläche entspricht. Im GKS-3D wird damit auch der Arbeitsbereich eines Eingabegerätes bezeichnet, zum Beispiel eines Digitalisierers.

4.37 echo — Echo

Die sofortige Anzeige des aktuellen Wertes eines Eingabegerätes für den Bediener des Bildschirmarbeitsplatzes.

4.38 echo area; echo volume — Echofeld; Echokörper

Ein Feld oder Körper in Gerätekoordinaten, der für die Darstellung einer Aufforderung oder eines Echos dient.

4.39 echo type — Echotyp

Ein Parameter für die Geräteinitialisierung, der die Echo-technik für eine bestimmtes logisches Eingabegerät auswählt.

4.40 edge — Kante

Die Menge der Randlinien der Polygone, die in einer Füllgebietsmenge definiert sind.

4.41 edge bundle table — Kantenbündeltabelle

Eine Tabelle, die bestimmte Werte für die geräteabhängigen Aspekte einer Füllgebietsmenge mit einem Füllgebietsindex verbindet. In GKS-3D enthält diese Tabelle die Einträge Kantenanzeiger, Kantentyp, Kantenbreitfaktor und Farbindex.

4.42 edge flag — Kantenanzeiger

Ein Aspekt der Füllgebietsmenge, der die Darstellung der Kante ein- und ausschaltet.

4.43 edge type — Kantentyp

Ein Aspekt von Füllgebietsmenge, der die Art der Kanten angibt.

4.44 edgewidth scale factor — Kantenbreitfaktor

Ein Aspekt von Füllgebietsmenge, der die relative Breite der Darstellung von Kanten angibt. Der Kantenbreitfaktor wird auf einen geräteabhängigen Nominalwert angewendet.

4.45 escape — Fluchtfunktion

Eine GKS-3D-Funktion zum Zugriff auf implementierungs- oder geräteabhängige Eigenschaften, die anderweitig nicht durch GKS-3D unterstützt werden. Eine Fluchtfunktion kann nicht zur Erzeugung graphischer Ausgabe benutzt werden.

4.46 event mode — Ereignismodus

Eine Betriebsart für logische Eingabegeräte, in welchem asynchrone Eingabe als Ereignisbericht in der Ereigniswarteschlange abgelegt wird, wenn der Auslöser betätigt wird.

4.47 event queue — Ereigniswarteschlange

Eine zeitlich geordnete Sammlung von Ereignisberichten.

4.48 event report — Ereignisbericht

Ein Eintrag in der Ereigniswarteschlange, der aus einem logischen Eingabewert und der Kennzeichnung des betreffenden logischen Eingabegerätes besteht.

4.49 feedback — Rückkopplung

Eine Anzeige für den Bediener, die ihm die Interpretation des logischen Eingabewertes durch das Anwendungsprogramm anzeigt.

4.50 fill area — Füllgebiet

Ein GKS-3D-Darstellungselement, das aus einem einzelnen Polygon besteht.

4.51 fill area bundle table — Füllgebietsbündeltabelle

Eine Tabelle, die bestimmte Werte für alle geräteabhängigen Aspekte des Füllgebiets-Darstellungselements mit dem Füllgebiets-Bündelindex verknüpft. Im GKS-3D enthält diese Tabelle Einträge, die aus Ausfüllung, Ausfüllungsindex und Farbindex bestehen.

4.52 fill area set — Füllgebietsmenge

Ein GKS-3D-Darstellungselement, das aus einer Menge von Füllgebieten einschließlich Kanten besteht.

4.53 front plane — Vordere Klippebene

Eine Ebene parallel zur Projektionsebene, dessen Lage als N-Koordinate im Referenzkoordinatensystem der Darstellungstransformation gegeben wird. Darstellungselemente vor dieser Ebene liegen außerhalb des Darstellungskörpers.

4.54 Generalized Drawing Primitive (GDP) — Verallgemeinertes Darstellungselement

Ein Darstellungselement, das es erlaubt, besondere geometrische Fähigkeiten eines graphischen Arbeitsplatzes wie z.B. Kurveninterpolation anzusprechen.

4.55 GKS-3D level — GKS-3D-Leistungsstufe

Zwei Werte im Bereich 0 bis 2 und a bis c, die zusammen eine Menge des funktionalen Leistungsumfangs definieren. Eine Implementierung des GKS-3D sieht genau die Funktionen einer Leistungsstufe vor.

4.56 GKS-3D Metafile (GKSM) — GKS-3D-Bilddatei

Eine sequentielle Datei, die vom GKS-3D geschrieben oder gelesen werden kann. Sie wird zur Langzeitspeicherung (und zur Weitergabe und Übertragung) von graphischer Information verwendet.

4.57 halfline — Mittellänge

Eine horizontale Linie im Zeichenkörper (siehe Bild 3), die die Mitte zwischen der Versalhöhe und der Schriftlinie bei einer horizontalen Zeichenfolge angibt. Alle Mittellängen einer Schriftart haben dieselbe Position innerhalb eines Zeichenkörpers.

4.58 hatch interior style — Schraffurausfüllung

Eine mögliche Methode zum Füllen des Inneren eines Füllgebiets und einer Füllgebietsmenge. Das Innere wird mit einem Muster paralleler oder kreuzender Schraffurlinien gefüllt, das aus der Schraffurtabelle des graphischen Arbeitsplatzes ausgewählt wird.

4.59 hatch table — Schraffurtabelle

Eine Tabelle von Schraffurwerten, die an einem Arbeitsplatz definiert sind.

4.60 hidden line/surface removal — Unterdrücken verdeckter Linien/Flächen

Unterdrücken der Teile von Darstellungselementen, die von anderen Darstellungselementen verdeckt werden.

4.61 highlighting — Hervorheben

Ein geräteunabhängiges Verfahren, um ein Segment durch Ändern seiner Sichtbarkeitsattribute besonders zu betonen. (Eine Verallgemeinerung des Blinkens.)

4.62 HLS — HLS

Eine Abkürzung für das Farbmodell Ton-Helligkeit-Sättigung.

4.63 hollow interior style — Ausfüllung leer

Eine mögliche Beschreibung des Inneren eines Füllgebiets und einer Füllgebietsmenge. Das Bild ist die Randlinie allein, einschließlich der Ränder, die beim Klippen entstehen.

4.64 HSV — HSV

Eine Abkürzung für das Farbmodell Ton-Sättigung-Intensität.

4.65 image — Bild

Das Erscheinen von Objekten nach dem Darstellungsprozeß.

4.66 implementation mandatory — Pflichtanforderung für die Implementierung

Beschreibung der Eigenschaften, die bei allen graphischen Arbeitsplätzen in allen Implementierungen der Norm identisch sein müssen.

4.67 input class — Eingabeklasse

Eine Menge von Eingabegeräten, die bezüglich ihrer Funktion gleichwertig sind. Im GKS-3D sind die Eingabeklassen: Auswähler, Liniengeber, Lokalisierer, Picker, Textgeber und Wertgeber.

4.68 input mode — Eingabemodus

Eine von drei möglichen Methoden, Daten von logischen Eingabegeräten zu erhalten: Anforderung, Abfrage, Ereignis.

4.69 inquiry function — Erfragefunktion

Eine GKS-3D-Funktion, die den Zweck hat, abhängig vom aktuellen Zustand des GKS-3D oder von einigen festen Eigenschaften der GKS-3D-Implementierung, Werte zurückzugeben. Es besteht keine Wirkung auf den Zustand des GKS-3D oder die graphische Darstellung.

4.70 interior style — Ausfüllung

Ein Aspekt, der angibt, wie das Innere eines Füllgebiets oder einer Füllgebietsmenge ausgefüllt wird.

4.71 linetype — Linientyp

Ein Aspekt, der die Art des Bildes von Linienzug wie durchgezogen, gestrichelt oder punktiert, angibt.

**4.72 linewidth scale factor —
Linienbreitfaktor**

Ein Aspekt, der die relative Breite des Bildes von Linienzug angibt. Der Linienbreitfaktor wird auf einen geräteabhängigen Nominalwert angewendet.

4.73 locator device — Lokalisierer

Ein logisches GKS-3D-Eingabegerät, das eine Position in Weltkoordinaten und die Nummer der Normierungstransformation liefert.

**4.74 logical input device —
Logisches Eingabegerät**

Eine Abstraktion eines oder mehrerer physischer Geräte, die logische Eingabewerte an das Programm übergeben.

**4.75 logical input value —
Logischer Eingabewert**

Der Eingabewert, den ein logisches Eingabegerät liefert.

4.76 marker — Marke

Ein Bezeichner zur Kennzeichnung einer bestimmten Position. Die Gestalt der Marke wird nicht transformiert.

**4.77 marker size scale factor —
Markenvergrößerungsfaktor**

Ein Aspekt, der die relative Größe des Bildes einer Marke angibt. Der Markenvergrößerungsfaktor wird auf einen geräteabhängigen Nominalwert angewendet.

4.78 marker type — Markentyp

Ein Aspekt, der die Art des Druckzeichens angibt, der zur Markendarstellung benutzt wird.

4.79 measure — Maßwert

Ein Wert, der durch ein oder mehrere physische Eingabegeräte und einer Abbildung ihrer Werte bestimmt ist. Die Form des Wertes wird durch eines der logischen Eingabegeräte bestimmt.

4.80 measure process — Maßwertprozeß

Ein Prozeß, der entsteht, wenn ein logisches Eingabegerät zur Interaktion bereit gemacht wird. Der aktuelle Zustand des Maßwertprozesses ist der Maßwert.

4.81 message — Nachricht

Eine Zeichenfolge, die vom Anwendungsprogramm an einen graphischen Arbeitsplatz geschickt wird, hauptsächlich zur Kommunikation mit dem Bediener.

4.82 MI — BE

Eine Abkürzung für die GKS-3D-Bilddateieingabe, eine Kategorie von graphischem Arbeitsplatz.

4.83 MO — BA

Eine Abkürzung für die GKS-3D-Bilddateiausgabe, eine Kategorie von graphischem Arbeitsplatz.

**4.84 normalization transformation, viewing transformation, window-to-viewport transformation — Normierungs-
transformation, Fenstertransformation**

Eine Transformation, die Rand und Inneres eines Fensters auf Rand und Inneres eines Darstellungsfeldes abbildet. Im GKS-3D bildet diese Transformation Positionen in Weltkoordinaten auf normierte Koordinaten ab.

**4.85 normalized device coordinates (NDC) —
Normierte Koordinaten**

Eine Koordinate in einem geräteunabhängigen Zwischenkoordinatensystem, das auf einen Bereich — häufig auf 0 bis 1 — normiert ist. Im GKS-3D können während eines Zwischenzustandes die Koordinaten außerhalb des angegebenen Bereichs liegen; durch das Klippen wird jedoch sichergestellt, daß die Ausgabe den Koordinatenbereich $[0,1] \times [0,1] \times [0,1]$ nicht überschreitet.

**4.86 normalized projection coordinates (NPC) —
Normierte Projektionskoordinaten**

Das geräteabhängige dreidimensionale kartesische Koordinatensystem, in dem die Komposition von Bildern für das graphische System spezifiziert wird. Das Projektionsdarstellungsfeld und das Gerätefenster werden in NPC spezifiziert.

4.87 oblique view — schiefe Projektion

Eine Parallelprojektion, bei der die Projektionsstrahlen nicht senkrecht zur Projektionsebene sind, oder eine perspektivische Projektion, bei der der Projektionsstrahl durch den Mittelpunkt des Projektionsfensters nicht senkrecht zur Projektionsebene ist.

4.88 operator — Bediener

Die Person, die physische Eingabegeräte bedient und insbesondere den Maßwert der logischen Eingabegeräte ändert und deren Auslöser betätigt.

**4.89 output primitive, graphic primitive,
display element — Darstellungselement**

Ein graphisches Grundelement, das zum Aufbau einer graphischen Darstellung verwendet werden kann.

**4.90 parallel transformation —
Parallelprojektion**

Die Transformation eines Objekts, bei der parallele Linien im Objekt auch parallel im resultierenden Bild sind, unabhängig der relativen Entfernung oder Tiefe.

**4.91 perspective transformation —
Perspektivische Abbildung**

Die Transformation eines Objekts, bei der parallele Linien im Objekt, die die Projektionsebene durchdringen, sich im resultierenden Bild in Abhängigkeit von relativer Entfernung und Tiefe sich gegenseitig annähern.

4.92 pick device — Picker

Ein logisches GKS-3D-Eingabegerät, das die mit dem Darstellungselement verknüpfte Pickerkennzeichnung und den zugehörigen Segmentnamen liefert.

4.93 pick identifier — Pickerkennzeichnung

Ein Kennzeichen, das mit einzelnen Darstellungselementen innerhalb eines Segments verknüpft ist und durch den Picker zurückgegeben wird. Die gleiche Pickerkennzeichnung kann an verschiedene Darstellungselemente vergeben werden.

4.94 pixel, picture element — Pixel, Bildelement

Das kleinste Element einer Darstellungsfläche, dem unabhängig eine Farbe oder Intensität zugeordnet werden kann.

4.95 polyline — Linienzug

Ein GKS-3D-Darstellungselement, das aus einer Anzahl verbundener Strecken zwischen gegebenen Punkten besteht.

4.96 polyline bundle table — Linienzugbündeltabelle

Eine Tabelle, die spezifische Werte für alle geräteabhängigen Aspekte des Linienzug-Darstellungselements mit dem Linienzugbündelindex verknüpft. Im GKS-3D enthält diese Tabelle Einträge, die aus Linientyp, Linienbreitfaktor und Farbindex bestehen.

4.97 polymarker — Polymarke

Ein GKS-3D-Darstellungselement, das aus einem Satz von Positionen besteht, die alle durch Marken gekennzeichnet werden.

4.98 polymarker bundle table — Polymarkenbündeltabelle

Eine Tabelle, die spezifische Werte für alle geräteabhängigen Aspekte des Polymarken-Darstellungselements mit dem Polymarkenbündelindex verknüpft. Im GKS-3D enthält diese Tabelle Einträge, die aus Markentyp, Markengrößefaktor und Farbindex bestehen.

4.99 primitive attribute — Darstellungsattribut

Ein Attribut für Darstellungselemente, das durch das Anwendungsprogramm auf geräteunabhängige Art ausgewählt wird, jedoch geräteabhängige Wirkungen haben kann.

4.100 projection plane — Projektionsebene

Siehe view plane — Darstellungsebene.

4.101 projection reference point (PRP) — Projektionsreferenzpunkt

Ein Punkt im Referenzkoordinatensystem der Darstellungstransformation, der die Richtung (bei Parallelprojektion) oder den Ausgangspunkt (bei perspektivischer Abbildung) der Projektionsstrahlen festlegt.

4.102 projector — Projektionsstrahlen

Eine konzeptuelle Linie, die durch jeden Punkt des Objekts geht und die Projektionsebene durchdringt.

4.103 prompt — Aufforderung

Eine Ausgabe, die dem Bediener anzeigt, daß ein bestimmtes logisches Eingabegerät verfügbar ist.

4.104 raster graphics — Rastergraphik

Eine Art graphischer Datenverarbeitung, bei der das Bild aus einer Matrix von Pixel zusammengesetzt wird, die in Zeilen und Spalten angeordnet sind.

4.105 RGB — RGB

Eine Abkürzung für das Farbmodell Rot-Grün-Blau.

4.106 rotation — Rotation

Drehen von Bildern oder Teilbildern um eine Achse. Im GKS-3D wird diese Fähigkeit nur auf Segmente angewandt.

4.107 scaling, zooming — Skalieren, Zoomen

Vergrößern oder Verkleinern von Bildern oder Teilbildern durch Multiplizieren der Koordinaten der Darstellungselemente mit einem konstanten Faktor. Im GKS-3D ist diese Operation durch Segment- und Darstellungstransformation verfügbar.

Anmerkung: Zum unterschiedlichen Skalieren in zwei orthogonale Richtungen werden zwei Faktoren benötigt.

4.108 segment — Segment

Eine Menge von Darstellungselementen, die als Einheit behandelt werden kann.

4.109 segment attribute — Segmentattribut

Ein Attribut, das nur Segmente betrifft. Im GKS-3D gibt es die Segmentattribute Sichtbarkeit, Hervorheben, Ansprechbarkeit, Segmentpriorität und Segmenttransformation.

4.110 segment priority — Segmentpriorität

Ein Segmentattribut, das bestimmt, welches von überlappenden Segmenten Vorrang für die graphische Ausgabe und Eingabe hat.

4.111 segment transformation — Segmenttransformation

Eine Transformation, die die Darstellungselemente im Segment an verschiedenen Positionen (Translation), in veränderter Größe (Skalieren) oder veränderter Richtung (Rotation) auf der Darstellungsfläche erscheinen läßt.

4.112 string — Zeichenfolge

Eine geordnete Menge von Zeichenkodierungen nach ISO 2022.

4.113 string device — Textgeber

Ein logisches GKS-3D-Eingabegerät, das als Ergebnis einen Text liefert.

4.114 stroke device — Liniengeber

Ein logisches GKS-3D-Eingabegerät, das eine Folge von Punkten einer Linie in Weltkoordinaten und die zugehörigen Indizes der Normierungs- und Darstellungstransformation liefert.

4.115 text — Text

Ein GKS-3D-Darstellungselement, das aus einer Zeichenfolge besteht, die an einer spezifizierten Position in einer spezifizierten Textebene dargestellt wird.

4.116 text alignment — Textausrichtung

Ein Aspekt von Text, der die Art der Ausrichtung spezifiziert. Dieser Aspekt hat eine horizontale und eine vertikale Komponente.

4.117 text bundle table — Textbündeltabelle

Eine Tabelle, die spezifische Werte für alle geräteabhängigen Aspekte des Text-Darstellungselements mit dem Textbündelindex verknüpft. Im GKS-3D enthält diese Tabelle Einträge, die aus Schriftart und -qualität, Zeichenbreitefaktor, Zeichenabstand und Farbindex bestehen.

4.118 text direction vectors — Textvektoren

Zwei Vektoren, die als Teil der Textfunktion spezifiziert werden und zusammen mit der Textposition die Textebene aufspannen.

4.119 text extent — Textausdehnung

Das kleinste Parallelogramm, das die Buchstaben einer Zeichenfolge vollständig umschließt.

4.120 text font and precision — Schriftart und Schriftqualität

Ein Aspekt des Text-Darstellungselements im GKS-3D mit den zwei Komponenten Schriftart und Schriftqualität, die zusammen die Form der Zeichen für die Ausgabe an einem bestimmten graphischen Arbeitsplatz festlegen. Zusätzlich schreibt die Schriftqualität die Genauigkeit vor, mit welcher die anderen Textaspekte mit der vom Anwendungsprogramm geforderten übereinstimmen. Die Schriftqualitäten werden in der Reihenfolge zunehmender Genauigkeit mit Lesbar Zeichen und Strich bezeichnet.

4.121 text path — Schreibrichtung

Ein Aspekt von Text, der die relative Positionierung von aufeinanderfolgenden Buchstaben in einer Zeichenfolge angibt.

4.122 text plane — Textebene

Die Ebene, in der ein Textdarstellungselement erscheint.

4.123 text position — Textposition

Ein Punkt, der als Teil der Textfunktion spezifiziert wird und der, gemeinsam mit den Textvektoren, die Textebene festlegt. Er bestimmt zusammen mit der Textausrichtung die Platzierung der Zeichenfolge.

4.124 translation, shift — Translation, Verschieben

Gleichmäßige Verschiebung der Position eines Bildes oder Teilbildes. Im GKS-3D ist dies nur für Segmente möglich.

4.125 trigger — Auslöser

Ein physisches Eingabegerät (oder mehrere Geräte), das der Bediener zu einem bestimmten maßgeblichen Zeitpunkt betätigt, um diesen Zeitpunkt dem System anzuzeigen.

4.126 valuator device — Wertgeber

Ein logisches GKS-3D-Eingabegerät, das eine reelle Zahl liefert.

4.127 view index — Darstellungsindex

Das Attribut für alle Darstellungselemente, das die Beschreibung der Darstellungstransformation auswählt, die auf Darstellungselemente angewendet wird.

4.128 view mapping matrix — Abbildungsmatrix der Darstellungstransformation

Eine 4×4-Matrix, die die Abbildung von Punkten vom Referenzkoordinatensystem der Darstellungstransformation in normierte Projektionskoordinaten spezifiziert.

4.129 view orientation matrix — Orientierungsmatrix der Darstellungstransformation

Eine 4×4-Matrix, die die Abbildung von Punkten von normierten Koordinaten in das Referenzkoordinatensystem der Darstellungstransformation spezifiziert.

4.130 view plane — Darstellungsebene

Eine Ebene parallel zur U-V-Ebene im Referenzkoordinatensystem der Darstellungstransformation (VRC). Sie wird durch einen N-Koordinatenwert im VRC spezifiziert.

4.131 view plane distance — Abstand der Darstellungsebene

Der Abstand der Darstellungsebene vom Referenzpunkt der Darstellungstransformation, gemessen entlang der Normalen der Darstellungsebene im Referenzkoordinatensystem der Darstellungstransformation.

4.132 view plane normal — Normale der Darstellungsebene

Ein Vektor in NDC oder Weltkoordinaten relativ zum Referenzpunkt der Darstellungstransformation, der die N-Koordinatenachse im Referenzkoordinatensystem der Darstellungstransformation definiert.

4.133 view reference coordinate system (VRC) — Referenzkoordinatensystem der Darstellungstransformation

Ein geräteunabhängiges dreidimensionales kartesisches Koordinatensystem, in dem die Parameter für die Darstellungstransformation spezifiziert werden. Die Achsen des Referenzkoordinatensystems der Darstellungstransformation werden mit U, V und N bezeichnet. Die Position und die Orientierung dieses Koordinatensystems werden, relativ zu Weltkoordinaten, durch den Referenzpunkt der Darstellungstransformation, die Normale der Darstellungsebene und dem Aufwärtsvektor der Darstellungstransformation definiert. Die Einheitsvektoren dieses Koordinatensystems haben gleiche Länge wie die des NDC-Systems.

4.134 view reference point — Referenzpunkt der Darstellungstransformation

Ein Punkt in NDC oder Weltkoordinaten, der den Ursprung des Referenzkoordinatensystems der Darstellungstransformation spezifiziert.

4.135 view table — Darstellungstabelle

Die geräteabhängige Darstellungstabelle auf jeden graphischen Arbeitsplatz. Es wird durch den geräteunabhängigen Darstellungsindex angesprochen.

4.136 view up vector — Aufwärtsvektor der Darstellungstransformation

Ein Vektor in NDC oder Weltkoordinaten. Die V-Achse des Referenzkoordinatensystems der Darstellungstransformation wird als orthogonale Projektion des Aufwärtsvektors auf die Ebene durch den Referenzpunkt und senkrecht zur Normalen der Darstellungsebene spezifiziert. Vektoren, die parallel zum Aufwärtsvektor in NDC oder Weltkoordinaten sind, erscheinen vertikal im endgültigen Bild.

4.137 view volume — Darstellungskörper

Der Darstellungskörper wird festgelegt durch das Darstellungsfenster, die vordere und hintere Klippebene und die Projektionsstrahlen durch die Seiten des Darstellungsfensters. Wenn der Projektionstyp PERSPEKTIVISCH ist, gehen alle Projektionsstrahlen durch den Referenzpunkt der Darstellungstransformation und der Darstellungskörper bildet einen Kegelstumpf. Wenn der Projektionstyp PARALLEL ist, sind die Projektionsstrahlen parallel zum Vektor, der den Referenzpunkt der Darstellungstransformation mit dem Mittelpunkt des Darstellungsfensters verbindet, und der Darstellungskörper bildet ein Parallelepiped. In zweidimensionalen Anwendungen reduziert sich der Darstellungskörper auf das Darstellungsfenster.

4.138 view window — Darstellungsfenster

Ein Rechteck in der Darstellungsebene. Projektionsstrahlen, die durch die Ecken des Darstellungsfensters gehen, definieren die linke, rechte, untere und obere Fläche des Darstellungskörpers.

4.139 viewing transformation — Darstellungstransformation

Eine Transformation, die Positionen in NDC-Koordinaten auf Positionen in NPC-Koordinaten abbildet.

4.140 viewport — Darstellungsfeld

Ein im Anwendungsprogramm definierter Teil des normierten Koordinatenbereichs. Im GKS-3D wird dieser Begriff nur für rechteckige Parallelepipede im normierten Koordinatenbereich verwendet, mit deren Hilfe die Normierungstransformation definiert wird.

4.141 window — Fenster

Ein definierter Teil eines Koordinatenraumes. Im GKS-3D wird dieser Begriff nur für rechteckige Parallelepipede im Weltkoordinatenbereich verwendet, mit deren Hilfe die Normierungstransformation definiert wird.

4.142 workstation — Graphischer Arbeitsplatz

Das GKS-3D ist auf der Konzeption abstrakter graphischer Arbeitsplätze aufgebaut. Sie stellen die funktionale Schnittstelle bereit, über die das Anwendungsprogramm die physischen Geräte steuert.

4.143 workstation dependent segment storage (WDSS) — Arbeitsplatzabhängiger Segmentspeicher

Segmentspeicher eines graphischen Arbeitsplatzes, der für die graphische Ausgabe verwendet wird. Die Segmente können nicht vom WDSS zu anderen Arbeitsplätzen übertragen werden.

4.144 workstation independent segment storage (WISS) — Arbeitsplatzunabhängiger Segmentspeicher

Spezieller Typ eines graphischen Arbeitsplatzes, mit dessen Hilfe Segmente gespeichert und später zu anderen graphischen Arbeitsplätzen übertragen werden können.

4.145 workstation mandatory — Pflichtanforderung für einen Arbeitsplatz

Beschreibung der Eigenschaften, die bei der Implementierung der Norm an allen graphischen Arbeitsplätzen identisch erfüllt sein müssen.

4.146 workstation transformation — Gerätetransformation

Eine Transformation, die das Gerätefenster auf das Gerätedarstellungsfeld abbildet, wobei X- und Y-Koordinaten gleich skaliert werden, aber nicht notwendigerweise auch die Z-Koordinaten.

4.147 workstation viewport — Gerätedarstellungsfeld

Ein rechteckiges Parallelepiped in Gerätekoordinaten. Alle graphische Ausgabe geht in dieses Feld.

4.148 workstation window — Gerätefenster

Ein rechteckiges Parallelepiped im normierten Projektionskoordinatensystem. Der Inhalt des Gerätefensters wird durch die Gerätetransformation in das Gerätedarstellungsfeld abgebildet.

4.149 world coordinates (WC) — Weltkoordinaten

Eine Ortsangabe in einem geräteunabhängigen kartesischen Koordinatensystem, das vom Anwendungsprogramm zur Beschreibung der Eingabe und Ausgabe verwendet wird.

Zitierte Normen

Siehe Abschnitt 2

Internationale Patentklassifikation

G 06 F 15/62

INTERNATIONAL STANDARD

**ISO
8805**

First edition
1988-09-15



INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ

Information processing systems — Computer graphics — Graphical Kernel System for Three Dimensions (GKS-3D) functional description

*Systèmes de traitement de l'information — Infographie — Système graphique de base pour
trois dimensions (GKS-3D) description fonctionnelle*

Reference number
ISO 8805 : 1988 (E)

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75 % approval by the member bodies voting.

International Standard ISO 8805 was prepared by Technical Committee ISO/TC 97, *Information processing systems*.

Users should note that all International Standards undergo revision from time to time and that any reference made herein to any other International Standard implies its latest edition, unless otherwise stated.

Contents

0	Introduction	1
1	Scope and field of application	4
2	References	5
3	Definitions	6
4	The Graphical Kernel System for Three Dimensions	14
4.1	About this International Standard	14
4.1.1	Specification	14
4.1.2	Registration	14
4.2	Introduction to GKS-3D	15
4.3	Concepts	17
4.4	Graphical output	20
4.4.1	Output primitives	20
4.4.2	Output primitive attributes	21
4.4.3	Polyline attributes	26
4.4.4	Polymarker attributes	27
4.4.5	Text attributes	27
4.4.6	Text extent and concatenation	39
4.4.7	Fill area attributes	39
4.4.8	Fill area set attributes	41
4.4.9	Cell array attributes	42
4.4.10	Generalized Drawing Primitive attributes	42
4.4.11	Colour	42
4.4.12	View index	43
4.4.13	Hidden line/hidden surface removal (HLHSR) identifier	43
4.5	Workstations	44
4.5.1	Workstation characteristics	44
4.5.2	Selecting a workstation	45
4.5.3	Deferring picture changes	46
4.5.4	Clearing the display space	50
4.5.5	Elimination of primitives outside segments	50
4.5.6	Sending messages to a workstation	51
4.5.7	Hidden line/hidden surface removal	51
4.6	Coordinate systems and transformations	52
4.6.1	Normalization transformations	52
4.6.2	Clipping	53
4.6.3	Workstation transformations	53
4.6.4	Transformation of locator input	56
4.6.5	Transformation of stroke input	59
4.6.6	Viewing	59
4.6.7	Viewing utility functions	61
4.7	Segments	70
4.7.1	Introduction to segments	70
4.7.2	Segment attributes	71
4.7.3	Segment transformations	72
4.7.4	Clipping and WDSS	72

4.7.5	Workstation Independent Segment Storage	72
4.7.6	WISS functions and clipping	73
4.8	Graphical input	74
4.8.1	Introduction to logical input devices	74
4.8.2	Logical input device model	75
4.8.3	Operating modes of logical input devices	76
4.8.4	Measures of each input class	79
4.8.5	Input queue and current event report	80
4.8.6	Initialization of input devices	81
4.8.7	Locator and stroke input using 2D input devices	82
4.9	GKS-3D Metafile interface	83
4.10	GKS-3D levels	85
4.10.1	Introduction to levels	85
4.10.2	The level structure	85
4.10.3	Level functionality	86
4.11	States of GKS-3D and inquiry functions	91
4.11.1	Description of states	91
4.11.2	Inquiry functions	92
4.12	Error handling	94
4.13	Special interfaces between GKS-3D and the application program	96
4.14	2D functions	97
5	GKS-3D functions	101
5.1	Notational conventions	101
5.2	Control functions	102
5.3	Output functions	111
5.4	Output attributes	125
5.4.1	Workstation independent primitive attributes	125
5.4.2	Workstation attributes (representations)	140
5.5	Transformation functions	149
5.5.1	Normalization transformation	149
5.5.2	View transformation	152
5.5.3	Workstation transformation	154
5.6	Segment functions	158
5.6.1	Segment manipulation functions	158
5.6.2	Segment attributes	163
5.7	Input functions	167
5.7.1	Initialization of input devices	167
5.7.2	Setting the mode of input devices	182
5.7.3	Request input functions	185
5.7.4	Sample input functions	190
5.7.5	Event input functions	195
5.8	Metafile functions	200
5.9	Inquiry functions	202
5.9.1	Introduction to inquiry functions	202
5.9.2	Inquiry function for operating state value	202
5.9.3	Inquiry functions for GKS-3D description table	202
5.9.4	Inquiry functions for GKS-3D state list	204
5.9.5	Inquiry functions for workstation state list	214
5.9.6	Inquiry functions for workstation description table	245
5.9.7	Inquiry functions for segment state list	275
5.9.8	Pixel inquiries	276
5.9.9	Inquiry function for GKS-3D error state list	279
5.10	Utility functions	280
5.11	Error handling	284
6	GKS-3D data structures	286
6.1	Notation and data types	286
6.2	Operating state	289

6.3	GKS-3D description table	290
6.4	GKS-3D state list	291
6.5	Workstation state list	294
6.6	Workstation description table	298
6.7	Segment state list	303
6.8	GKS-3D error state list	304
A	Function lists	305
A.1	Alphabetic	305
A.2	Order of appearance	310
A.2.1	Control functions 5.2	310
A.2.2	Output functions 5.3	310
A.2.3	Output attributes 5.4	310
A.2.3.1	Workstation independent primitive attributes 5.4.1	311
A.2.3.2	Workstation attributes (representations) 5.4.2	311
A.2.4	Transformation functions 5.5	311
A.2.4.1	Normalization transformation 5.5.1	311
A.2.4.2	View transformation 5.5.2	312
A.2.4.3	Workstation transformation 5.5.3	312
A.2.5	Segment functions 5.6	312
A.2.5.1	Segment manipulation functions 5.6.1	312
A.2.5.2	Segment attributes 5.6.2	312
A.2.6	Input functions 5.7	312
A.2.6.1	Initialization of input devices 5.7.1	312
A.2.6.2	Setting the mode of input devices 5.7.2	313
A.2.6.3	Request input functions 5.7.3	313
A.2.6.4	Sample input functions 5.7.4	313
A.2.6.5	Event input functions 5.7.5	313
A.2.7	Metafile functions 5.8	313
A.2.8	Inquiry functions 5.9	314
A.2.8.1	Inquiry function for operating state value 5.9.2	314
A.2.8.2	Inquiry functions for GKS-3D description table 5.9.3	314
A.2.8.3	Inquiry functions for GKS-3D state list 5.9.4	314
A.2.8.4	Inquiry functions for workstation state list 5.9.5	314
A.2.8.5	Inquiry functions for workstation description table 5.9.6	315
A.2.8.6	Inquiry functions for segment state list 5.9.7	316
A.2.8.7	Pixel inquiries 5.9.8	316
A.2.8.8	Inquiry function for GKS-3D error state list 5.9.9	316
A.2.9	Utility functions 5.10	316
A.2.10	Error handling 5.11	316
A.3	Ordered by level	316
A.3.1	Level 0a	316
A.3.2	Level 0b	319
A.3.3	Level 0c	320
A.3.4	Level 1a	320
A.3.5	Level 1b	321
A.3.6	Level 1c	322
A.3.7	Level 2a	322
A.4	Applicability to workstation groups	323
B	Error list	329
B.1	Implementation dependent	329
B.2	States	329
B.3	Workstations	329
B.4	Transformations	329
B.5	Output attributes	330
B.6	Output primitives	330
B.7	Segments	331
B.8	Input	331

B.9	Metafiles	331
B.10	Escape	331
B.11	Miscellaneous	331
B.12	System	331
B.13	3D transformations	332
B.14	3D output attributes	332
B.15	3D output primitives	332
B.16	Reserved errors	332
C	Interfaces	333
C.1	General	333
C.2	Language binding	333
C.3	Implementation	334
D	Allowable differences in GKS-3D implementations	336
D.1	General	336
D.2	Global differences	336
D.3	Workstation dependent differences	337
E	Metafile structure	340
E.1	Metafiles	340
E.1.1	General	340
E.1.2	ISO 8632 Metafile	340
E.1.3	Metafile designed for GKS-3D	341
E.2	File format and data format	341
E.3	Generation of metafiles	342
E.4	Interpretation of metafiles	346
E.4.1	General	346
E.4.2	Control items	346
E.4.3	Output primitives	346
E.4.4	Output primitive attributes	346
E.4.5	Workstation attributes	346
E.4.6	Transformations	346
E.4.7	Segment manipulation	347
E.4.8	Segment attributes	347
E.4.9	Items from a GKSM written by a GKS system	347
E.5	Control items	347
E.6	Items for output primitives	350
E.7	Items for output primitive attributes	351
E.8	Items for workstation attributes	355
E.9	Items for transformations	357
E.9.1	Items for clipping	357
E.9.2	Items for viewing	357
E.9.3	Items for HLHSR	357
E.9.4	Items for workstation transformation	358
E.10	Items for segment manipulation	358
E.11	Items for segment attributes	358
E.12	User items	359
F	Sample programs	360
G	GKS-3D functions summary	369
G.1	Control functions	369
G.2	Output functions	369
G.3	Output attributes	370
G.3.1	Workstation independent primitive attributes	370
G.3.2	Workstation attributes (representations)	372
G.4	Transformation functions	372
G.4.1	Normalization transformation	372
G.4.2	Viewing transformation	372
G.4.3	Workstation transformation	373

Information processing systems - Computer graphics - Graphical Kernel System for Three Dimensions (GKS-3D) functional description

0 Introduction

The Graphical Kernel System for Three Dimensions (GKS-3D) provides a set of functions for:

- a) definition and display of 2D and 3D graphical data;
- b) storage and manipulation of graphical data;
- c) input of graphically related data.

The main reasons for introducing this International Standard for computer graphics is:

- d) to allow application programs involving graphics to be easily portable between different installations;
- e) to aid the understanding and use of graphics methods by application programmers;
- f) to serve manufacturers of graphics equipment as a guideline in providing useful combinations of graphics capabilities in a device.

In order to reach these main objectives, the GKS-3D design was based on the following requirements:

- g) GKS-3D should include all the capabilities that are essential for the whole spectrum of graphics, from simple passive output to highly interactive applications.
- h) The whole range of graphics devices, including vector and raster devices, microfilm recorders, storage tube displays, refresh displays and colour displays should be controllable by GKS-3D in a uniform way.
- i) GKS-3D should provide all the capabilities required by a majority of applications.

These requirements were used to formulate a number of principles that were used to judge specific design alternatives. Thus it was possible to contribute to the overall design goals while focussing on certain aspects. Five design aspects were identified, each having a group of principles

Introduction

j) Design goals: The following principles should not be violated by any technical design:

- 1) consistency: the mandatory requirements of GKS-3D should not be mutually contradictory;
 - 2) compatibility: other standards or commonly accepted rules of practice should not be violated;
 - 3) orthogonality: the functions or modules of GKS-3D should be independent of each other, or the dependency should be structured and well defined.
- k) Functional capabilities: The following principles were used to define the extent of GKS-3D:
- 1) completeness: all functions that a majority of applications want to use on a given level of functionality should be included;
 - 2) minimality: functions that are unnecessary for applications of a given level of functionality should not be provided;
 - 3) compactness: an application should be able to achieve a desired result by a set of functions and parameters that is as small as possible;
 - 4) richness: a rich set of functions offers an extensive range of facilities that stretches beyond the basic functions and includes higher order capabilities.

It is obvious that there is a trade off between the principles in this group. Therefore, the functions of GKS-3D are organized in nine levels. An implementation of GKS-3D provides precisely the functions of one of these levels. While the lowest level contains only a minimal set of functions, higher levels are allowed to extend beyond the basic needs towards greater richness.

l) User interface design: The following principles were used to define the user interface design:

- 1) user friendliness: GKS-3D should allow the design of a desirable user interface;
- 2) clarity: the concepts and functional capabilities of GKS-3D should be easily understandable, especially by the application programmer;
- 3) error handling: failure of system functions or modules, caused by errors of the system itself or by the application program, should be treated in such a way that the error reaction is clearly understandable and informative to the application programmer and that the impact on the system and the application program is as small as possible.

Clarity and sound error handling are essential parts of user friendliness. Error handling is an integral part of GKS-3D. To aid clarity, the system and its state can be presented to the user in an easily comprehensible manner.

Clarity applies not only to the system design but also to the system description. To this end, the GKS-3D specification is divided into a general description, a description of the underlying logical data structures representing the state of the system, and a description of the functions and their effects on these data structures.

m) Graphics devices: The following principles are associated with the range of graphics devices that can be addressed by GKS-3D:

- 1) device independence: GKS-3D functions should be designed to allow an application program, using these functions, to address facilities of quite different graphics output and input devices without modification of the program structure;
- 2) device richness: the full capabilities of a wide range of different graphics output and input devices should be accessible from the functions of GKS-3D.

These principles led to a fundamental concept underlying the GKS-3D architecture: the concept of multiple independent graphical workstations connected to and driven by GKS-3D. The application program can inquire the capabilities of every workstation. The GKS-3D design includes escape functions that are easily identifiable within an application program and can be used to access special facilities of a particular device.

Introduction

- n) Implementation: The last group of principles is related to the implementation of GKS-3D:
- 1) implementability: it should be possible to support the GKS-3D functions in most host languages, on most operating systems and with most graphics devices;
 - 2) language independence: it should be possible to access the facilities of GKS-3D from all International Standard programming languages;
 - 3) efficiency: GKS-3D should be capable of being implemented without time consuming algorithms;
 - 4) robustness: the operator and application programmer should be protected in the best possible way from hardware or software failure of the system.

The five groups of principles are interconnected. For example, design goals and functional capabilities both contribute to user friendliness. Efficiency is also important when considering response time in an interactive environment. Some principles may be conflicting, such as richness versus minimality, comprehensive error handling versus efficiency, and compactness versus device richness. Compromises needed to be made to achieve the overall design objective: GKS-3D should have an easily comprehensible structure and a set of functions that enables a vast majority of computer graphics users to design portable, device independent application programs addressing the whole range of computer graphics equipment.

o) Relationship to ISO 7942: This International Standard adheres to the following design goals with respect to ISO 7942:

- 1) the relationship between the functions in ISO 7942 and the functions described in this International Standard should be precisely defined and not be implementation dependent;
- 2) the format of functions and their parameters in ISO 7942 will remain unchanged in this International Standard;
- 3) additional functions will be added solely to support the new 3D functionality.

p) Compatibility of Application Programs: a major goal of this International Standard is that application programs written using the functions specified in ISO 7942 will run without modification on implementations conforming to this International Standard and produce the same results on workstations of the same type, subject to the allowable differences as detailed in Annex D. However, it is recognized that in certain situations, especially involving the use of metafiles, this goal cannot always be achieved (see Annex D for further details).

q) Relationship to ISO 9592: This International Standard has the following relationship to ISO 9592 (PHIGS):

The primary emphasis of PHIGS is different from that of GKS/GKS-3D. Unlike GKS/GKS-3D, PHIGS is not restricted to graphics and provides data structure definition and editing facilities which are not necessarily related to graphical output. PHIGS provides data structure facilities which can be rendered as graphical output, rather than being purely directed towards providing graphical output and interaction.

GKS/GKS-3D is oriented towards the direct output of graphical primitives grouped into segments, PHIGS provides data structures which are traversed to produce graphical output. However, the output primitives and their attributes generated at structure traversal time are almost identical to those of GKS-3D.

PHIGS provides the same underlying model of input as GKS-3D. The measure of the pick logical input device is extended to identify the structure element in the PHIGS Central Structure Store which generates the primitive picked by the operator.

Annexes C to H are given for information; they do not form part of the specification.

1 Scope and field of application

This International Standard specifies a set of functions for computer graphics programming, the Graphical Kernel System for Three Dimensions (GKS-3D). GKS-3D is a basic graphics system for applications that produce computer generated three dimensional pictures on graphics output devices. It supports operator input and interaction by supplying basic functions for graphical input and picture segmentation. It allows storage and dynamic modification of pictures. A fundamental concept in GKS-3D is the workstation, consisting of a number of input devices and a single output device. Several workstations can be used simultaneously. The application program is allowed to adapt its behaviour at a workstation to make best use of workstation capabilities. This International Standard includes functions for storage on and retrieval from an external graphics file. Also, the functions are organized in upward compatible levels with increasing capabilities.

This International Standard provides application programs with the capability to define and display 3D graphical primitives, specified using 3D coordinates. In addition, the GKS-3D input model provides 3D locator and stroke input.

The output primitives of this International Standard differ from the output primitives of ISO 7942 in the following ways:

- a) the polyline and polymarker primitives are defined by an arbitrary sequence of 3D coordinates;
- b) the text, fill area and cell array primitives remain planar, but may lie in a plane positioned arbitrarily in 3D space;
- c) the fill area set primitive is introduced.

Functions are provided in this International Standard to generate these primitives.

The functions of ISO 7942 can be used in this International Standard. However, the output functions generate instances of 3D primitives — instances that are constrained to lie in a plane. Furthermore, additional capabilities are provided in this International Standard to:

- d) specify views of 3D objects;
- e) obtain information from 3D input devices;
- f) permit workstations to perform hidden line/hidden surface removal (HLHSR).

This International Standard provides no specific functions for controlling such rendering techniques as light source, shading, texturing, and shadow computations. However, workstations may apply these techniques, locally as the primitives and attributes are realized, provided that the general guidelines of Annex D concerning allowable workstation differences are followed.

NOTE - For certain parameters of the functions, GKS-3D defines value ranges as being reserved for registration (see 4.1.2). The meanings of these values will be defined using the established procedures.

GKS-3D defines a language independent nucleus of a graphics system. For integration into a programming language, GKS-3D is embedded in a language dependent layer obeying the particular conventions of that language.

2 References

- ISO 646, *Information processing - ISO 7-bit coded character set for information interchange.*
- ISO 2022, *Information processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques.*
- ISO 2382/13, *Data processing - Vocabulary - Part 13: Computer graphics.*
- ISO 6093, *Information processing - Representation of numerical values in character strings for information interchange.*
- ISO 7942, *Information processing systems - Computer graphics - Graphical Kernel System (GKS) functional description*
- ISO 8632, *Information processing systems - Computer graphics - Metafile for the storage and transfer of picture description information*
 - Part 1 : Functional description.
 - Part 2 : Character encoding.
 - Part 3 : Binary encoding.
 - Part 4 : Clear text encoding.
- ISO/IEC 8806 *Information processing systems - Computer graphics - Graphical Kernel System for Three Dimensions (GKS-3D) Language Bindings*²⁾
 - Part 1 : FORTRAN
 - Part 2 : PASCAL
 - Part 3 : ADA
 - Part 4 : C
- ISO/IEC 9592, *Information processing systems - Computer graphics - The Programmer's Hierarchical Interactive Graphics System (PHIGS)*
 - Part 1 : Functional description.
 - Part 2 : Archive file format
 - Part 3 : Clear text encoding of archive file
- CIE Recommendations on colour space - supplement to CIE publication 15
- CIE 1976 *Supplementary standard colour metric of server and coordinate systems*

1) Two addenda are being produced which support segmentation and three dimensions.

2) At present at the stage of draft; publication anticipated in due course.

3 Definitions

For the purposes of this International Standard the following definitions apply.

NOTE: - As far as possible, commonly accepted graphics terminology is used.

3.1 acknowledgedgement: Output to the operator of a logical input device indicating that a trigger has fired.

3.2 aspect ratio: The ratio of lengths along the principal axes of an object.

3.3 aspect source flag (ASF): A flag indicating whether a particular workstation dependent aspect of a primitive is selected from an attribute bundle, or as an individual attribute selection.

3.4 aspects of primitives: Ways in which the appearance of a primitive can vary. Some aspects are controlled directly by primitive attributes, some are controlled indirectly through a bundle table. Primitives inside segments have an aspect controlled through the segment containing them, for example highlighting; primitives outside segments do not.

3.5 attribute: Attributes control the properties of output primitives and segments. There are four types of primitive attributes: geometric, non-geometric, viewing and identification. The geometric and non-geometric attributes control the values of aspects of primitives.

3.6 back plane: A plane parallel to the view plane whose location is specified as an N coordinate value in the view reference coordinate system. Primitives behind the back plane lie outside the view volume.

3.7 baseline: A horizontal line within a character body (see figure 3) which, for many character definitions, has the appearance of being a lower limit of the character shape. A descender passes below this line. All baselines in a font are in the same position in the character bodies.

3.8 bundle index: An attribute of an output primitive which is an index into a bundle table. It defines workstation dependent aspects of a primitive.

3.9 bundle table: A workstation dependent table specifying aspects of one or more output primitives. GKS-3D has polyline, polymarker, text, fill area, and edge bundle tables.

3.10 capline: A horizontal line within a character body (see figure 3) which, for many character definitions, has the appearance of being the upper limit of the character shape. An ascender may pass above this line and in some languages an additional mark (for example an accent) over the character may be defined above this line. All caplines in a font are in the same position in the character bodies.

3.11 cell array: An output primitive consisting of a parallelogram of equal sized cells, each of which is a parallelogram and has a single colour.

3.12 centre of projection: See Projection Reference Point (PRP).

3.13 centreline: A vertical line bisecting the character body (see figure 3).

3.14 character base vector: An aspect of text which defines the principal base direction of the text string. It is a two-dimensional vector in the text plane specified in the text output

3.15 character body: The rectangle defining the horizontal and vertical limits of an individual character.

3.16 character expansion factor: An aspect of text which specifies the deviation of character width from the defined nominal value.

3.17 character height: An aspect of text which specifies the nominal height of an upper case character.

3.18 character set: A registered interpretation for entries in the character code table (see ISO 2022).

Definitions

3.19 character spacing: An aspect of text which specifies the fraction of font nominal character height to be added between adjacent character bodies in a string.

3.20 character up vector: An aspect of text which defines the principal up direction of the text string. It is a two-dimensional vector in the text plane specified in the text output function.

3.21 character width: An aspect of text which specifies the nominal width of an upper case character.

3.22 choice device: A GKS-3D logical input device providing a positive integer defining one of a set of alternatives.

3.23 CIE: Abbreviation for the Commission Internationale de L'Eclairage. Used to refer to the CIE universal colour definition system used as a colour model.

3.24 CIE L*u*v* colour model: A colour model with an approximately uniform colour space for small colour differences and an associated colour difference formula. Its components are L* (Lightness) and u*, v* (Chromaticity). The coordinates of the associated chromaticity diagram are u* and v*.

3.25 CIE Tristimulus colour model: A colour model allowing the description of all colours within the colour gamut by non-negative tristimulus colour values X, Y, Z.

3.26 clipping: Removing those parts of output primitives which lie outside a specified volume.

3.27 clipping volume: Either the normalization viewport clipping volume, the view clipping volume or the intersection of the two. The normalization viewport clipping volume is a rectangular parallelepiped in normalized device coordinates (NDC). The view clipping volume is in the viewing reference coordinate system and is defined by the window, front and back planes, and projectors of the corners of the window. Output primitives within these volumes will not be clipped. Parts of output primitives on the planes forming the edges of these volumes are considered to be within the volume.

3.28 colour model: The method by which the application program describes a colour.

3.29 colour table: A workstation dependent table, in which the entries specify the values defining a particular colour.

3.30 coordinate graphics; line graphics: Computer graphics in which display images are generated from display commands and coordinate data.

3.31 device coordinates (DC): A device dependent coordinate system. In GKS-3D, DC units are metres on a device capable of producing a precisely scaled image, and appropriate workstation dependent units otherwise.

3.32 device driver: The device dependent part of a GKS-3D implementation that supports a physical graphics device. The device driver generates device dependent output and handles device dependent interaction.

3.33 device space: The space defined by the addressable points of a display device.

3.34 display device: A graphics device on which images can be represented. A display device is one possible component of a workstation.

3.35 display image; picture: A collection of output primitives or segments that are represented together at any one time on a display space.

3.36 display space: (1) That portion of the device space corresponding to the area available for displaying images. (2) The working space of an input device such as a digitiser.

3.37 echo: The immediate notification to the operator of the current measure of a logical input device.

3.38 echo area; echo volume: An area or volume, defined in device coordinates, which may be used for the display of a prompt or echo.

- Definitions**
- 3.39 echo type:** A parameter of device initialization which selects the echo technique for a particular logical input device.
- 3.40 edge:** The set of boundaries of the polygons defined in the fill area set primitive.
- 3.41 edge bundle table:** A table associating specific values for workstation dependent aspects of a fill area set primitive with an edge bundle index. In GKS-3D, this table contains entries consisting of edge flag, edgetype, edgewidth scale factor, and colour index.
- 3.42 edge flag:** An aspect of fill area set which enables or disables the display of edges.
- 3.43 edge type:** An aspect of fill area set which indicates the style of the edges.
- 3.44 edgewidth scale factor:** An aspect of fill area set which indicates the relative width of the image of an edge. The edgewidth scale factor is applied to a workstation dependent nominal value.
- 3.45 escape:** A function which provides access to implementation dependent or device dependent features, not concerned with the generation of graphical output.
- 3.46 event mode:** An operating mode for a logical input device in which asynchronous input is placed on the event queue as an event report when a trigger fires.
- 3.47 event queue:** A time-ordered collection of event reports.
- 3.48 event report:** An entry in the event queue which consists of a logical input value and identification of the logical input device responsible.
- 3.49 feedback:** Output indicating to the operator the application program's interpretation of a logical input value.
- 3.50 fill area:** An output primitive consisting of a single polygon.
- 3.51 fill area bundle table:** A table associating specific values for workstation dependent aspects of fill area and fill area set primitives with bundle table index values. In GKS-3D, this table contains entries for interior style, style index, and colour index. The workstation dependent aspects of the fill area set edge are contained in the edge bundle table.
- 3.52 fill area set:** An output primitive consisting of a set of fill areas with edge control.
- 3.53 front plane:** A plane parallel to the view plane which is specified as an N coordinate value in the view reference coordinate system. Primitives in front of the front plane lie outside the view volume.
- 3.54 Generalized Drawing Primitive (GDP):** An output primitive which accesses implementation dependent and workstation dependent geometrical capabilities such as curve drawing.
- 3.55 GKS-3D level:** Two values in the range 0 to 2 and a c which together define a set of functional capabilities of GKS-3D. An implementation of GKS-3D provides precisely the functions of one level.
- 3.56 GKS-3D Metafile (GKSM):** A sequential file that can be written or read by GKS-3D and is used for long-term storage (and for transmittal and transferral) of graphical information.
- 3.57 halffline:** A horizontal line between the capline and the baseline within the character body (see figure 3), about which a horizontal string of characters in a font would appear centrally placed in the vertical direction. All halfflines in a font are in the same position in the character bodies.
- 3.58 hatch interior style:** One possible representation of the interior of a fill area set or fill area primitive. The interior is filled with a pattern of parallel or crossing hatch lines, selected from the workstation hatch table.
- 3.59 hatch table:** The table of the hatch values defined on a workstation.
- 3.60 hidden line/hidden surface removal:** Removal of those parts of output primitives which are obscured by other output primitives.
- 3.61 highlighting:** A device independent way of emphasizing a segment by modifying its visual attributes.

Definitions

- 3.62 HLS:** An abbreviation for the Hue, Lightness, Saturation colour model.
- 3.63 hollow interior style:** One possible representation of the interior of a fill area or fill area set primitive. The image is the boundary line only, including any boundaries created by clipping.
- 3.64 HSV:** An abbreviation for the Hue, Saturation, Value colour model.
- 3.65 image:** The appearance of objects after rendering.
- 3.66 implementation mandatory:** Implementation mandatory describes a property that is required to be realized identically on all workstations of all implementations of GKS-3D.
- 3.67 input class:** The characterization of the functionality of a logical input device. There are six classes of logical input device: LOCATOR, STROKE, VALUATOR, CHOICE, PICK, and STRING.
- 3.68 input mode:** One of the three possible methods of obtaining data from a logical input device: REQUEST, SAMPLE, or EVENT.
- 3.69 inquiry function:** A mechanism for communicating to the application program data contained in a state list or description table.
- 3.70 interior style:** An aspect which indicates the style used to fill the interior of a fill area or fill area set.
- 3.71 linetype:** An aspect which indicates the style of the image of a polyline, such as solid, dashed or dotted.
- 3.72 linewidth scale factor:** An aspect which indicates the relative width of the image of a polyline. The linewidth scale factor is applied to a workstation dependent nominal value.
- 3.73 locator device:** A logical input device providing a position in world coordinates, a normalization transformation number, and a view index.
- 3.74 logical input device:** An abstraction of one or more physical input devices which delivers logical input values to the application program.
- 3.75 logical input value:** A value delivered by a logical input device.
- 3.76 marker:** A glyph with a specified appearance which is used to identify a location on a display surface. The shape of a marker is not subject to transformation.
- 3.77 marker size scale factor:** An aspect which indicates the relative size of the image of a marker. The marker size scale factor is applied to a workstation dependent nominal value.
- 3.78 marker type:** An aspect which selects the type of glyph used for markers.
- 3.79 measure:** A value which is determined by the values of one or more physical input devices and a mapping from these values to a form required by one of the logical input classes.
- 3.80 measure process:** A process which comes into existence when a logical input device is enabled for interaction. The current state of a measure process is the measure.
- 3.81 message:** A string sent by the application program to a workstation principally for communication with the operator.
- 3.82 MI:** An abbreviation for GKS-3D metafile input, a category of workstation.
- 3.83 MO:** An abbreviation for GKS-3D metafile output, a category of workstation.
- 3.84 normalization transformation; window-to-viewpoint transformation:** A transformation that maps the boundary and interior of a window to the boundary and interior of a viewport. In GKS-3D this transformation maps positions in world coordinates to normalized device coordinates.
- 3.85 normalized device coordinates (NDC):** A coordinate specified in a device independent intermediate coordinate system, normalized to some range, typically 0 to 1. In GKS-3D, during an intermediate state the coordinates may lie outside the defined range, but associated clipping information ensures that the output does not exceed the coordinate range $[0,1] \times [0,1] \times [0,1]$.

Definitions

3.86 normalized projection coordinates (NPC): The device-dependent three dimensional Cartesian coordinate system in which the composition of images is specified to the graphics system. The projection viewport and workstation window are specified in NPC space.

3.87 oblique view: A parallel projection whose projectors are not perpendicular to the view plane or a perspective projection where the projector through the centre of the view window is not perpendicular to the view plane.

3.88 operator: Person manipulating physical input devices so as to change the measures of logical input devices and cause their triggers to fire.

3.89 output primitive: A fundamental graphical display entity representing a geometric shape.

3.90 parallel transformation: The transformation of an object in which parallel lines in the object appear parallel in the resulting image without regard to relative distance or depth.

3.91 perspective transformation: The transformation of an object in which parallel lines in the object which intersect the view plane appear to converge in the resulting image as a function of relative distance or depth.

3.92 pick device: A GKS-3D logical input device providing the pick identifier attached to an output primitive and the associated segment name.

3.93 pick identifier: A name, attached to individual output primitives within a segment, and returned by the pick device. The same pick identifier can be assigned to different output primitives.

3.94 pixel; picture element: The smallest element of a display space that can be independently assigned a colour or intensity.

3.95 polyline: An output primitive consisting of a connected sequence of straight lines between specified points.

3.96 polyline bundle table: A table associating specific values for all workstation dependent aspects of a polyline primitive with a polyline bundle index. In GKS-3D, this table contains entries consisting of linetype, linewidth scale factor, and colour index.

3.97 polymarker: An output primitive consisting of a set of locations, each indicated by the same type of marker.

3.98 polymarker bundle table: A table associating specific values for all workstation dependent aspects of a polymarker primitive with a polymarker bundle index. In GKS-3D, this table contains entries consisting of marker type, marker size scale factor, and colour index.

3.99 primitive attribute: Primitive attribute values (for output primitives) are selected by the application in a workstation independent manner, but can have workstation dependent effects.

3.100 projection plane: See view plane.

3.101 projection reference point (PRP): A point in view reference coordinate space which determines the direction of projectors when the projection type is PARALLEL, or from which all projectors emanate when the projection type is PERSPECTIVE.

3.102 projector: A conceptual line which passes through each point of an object and intersects the view plane.

3.103 prompt: Output to the operator indicating that a specific logical input device is available.

3.104 raster graphics: Computer graphics in which a display image is composed of an array of pixels arranged in rows and columns.

3.105 RGB: An abbreviation for the Red, Green, Blue colour model.

3.106 rotation: Turning all or part of a display image about an axis. In GKS-3D, this capability is available via the segment and viewing transformations.

3.107 scaling; zooming: Enlarging or reducing all or part of a display image by multiplying the coordinates of display elements by a constant value. In GKS-3D, this capability is available via the

Definitions

segment and viewing transformations.

NOTE - For different scaling in three orthogonal directions three constant values are required.

3.108 segment: A collection of display elements that can be manipulated as a unit.

3.109 segment attributes: Attributes that apply only to segments. In GKS-3D, segment attributes are visibility, highlighting, detectability, segment priority, and segment transformation.

3.110 segment priority: A segment attribute used to determine which of several overlapping segments takes precedence for graphic output and input.

3.111 segment transformation: A transformation that causes the display elements defined by a segment to appear with varying position (translation), size (scale), and/or orientation (rotation) on the display space.

3.112 string: An ordered collection of character codes as described in ISO 2022.

3.113 string device: A logical input device providing a string.

3.114 stroke device: A logical input device providing a sequence of points in world coordinates and an associated normalization transformation number and view index.

3.115 text: An output primitive consisting of a character string drawn at a specified position on a specified text plane.

3.116 text alignment: An aspect of text which specifies the mode of justification. This aspect has horizontal and vertical components.

3.117 text bundle table: A table associating specific values for all workstation dependent aspects of a text primitive with a text bundle index. In GKS-3D, this table contains entries consisting of text font and precision, character expansion factor, character spacing and colour index.

3.118 text direction vectors: Two vectors, specified as part of the text output function, which, together with the text position, define the text plane.

3.119 text extent: A parallelogram defining the minimum area which completely encloses the character bodies in a string.

3.120 text font and precision: An aspect of text in GKS-3D, having two components, font and precision, which together determine the shape of the characters being output, on a particular workstation. In addition, the precision describes the fidelity with which the other text aspects match those requested by an application program. In order of increasing fidelity, the precisions are: STRING, CHAR and STROKE.

3.121 text path: An aspect of text which controls the relative positioning of successive characters in a string.

3.122 text plane: The plane in which a text primitive appears.

3.123 text position: A point, specified as part of the text output function, which, together with the text direction vectors, determine the text plane. It also determines, together with the text alignment, the location of the string.

3.124 translation; shift: The application of a constant displacement to the position of all or part of a display image. In GKS-3D, this capability is restricted to segments.

3.125 trigger: A physical input device or set of devices that an operator can use to indicate significant moments in time.

3.126 valuator device: A logical input device providing a real number.

3.127 view index: The attribute of all primitives that selects the viewing representation applied to the primitives at display time.

3.128 view mapping matrix: A 4×4 matrix which specifies the transformation of points from view reference coordinates to normalized projection coordinates.

Definitions

3.129 view orientation matrix: A 4×4 matrix which specifies the transformation of points from world coordinates to view reference coordinates.

3.130 view plane: A plane parallel to the U-V plane in view reference coordinate space (VRC). It is specified as an N coordinate value in VRC.

3.131 view plane distance: The distance of the view plane from the view reference point measured along the view plane normal vector in view reference system coordinates.

3.132 view plane normal: A vector in NDC or world coordinates, relative to the view reference point, which defines the N coordinate axis in view reference coordinate space.

3.133 view reference coordinate system (VRC): A device independent three-dimensional Cartesian coordinate system in which the parameters for the view mapping transformation are specified. The axes of the view reference coordinate system are called the U, V, and N axes. The position and orientation of this coordinate system, relative to world coordinates are defined by the view reference point, the view plane normal and the view up vector. The unit vectors of this system are of equal length to that of the NDC system.

3.134 view reference point: A point in NDC or world coordinates which defines the view reference coordinate system origin.

3.135 view table: The workstation dependent table of views found on each workstation. It is referenced by the workstation independent view index attribute.

3.136 view up vector: A vector defined in NDC or world coordinate space. The view reference coordinate system V coordinate axis is defined as an orthogonal projection of the view up vector onto the plane through the view reference point and perpendicular to the view plane normal. Vectors which are parallel to the view up vector in NDC or world coordinate space will appear vertical in the final image.

3.137 view volume: The view volume is determined by the view window defined in the view plane, the front plane, the back plane, and projectors through the corners of the view window. If the projection type is PERSPECTIVE, the projectors all pass through the projection reference point, the view volume is therefore a frustum. If the projection type is PARALLEL, the projectors are parallel to the vector joining the projection reference point to the centre of the view window, the view volume in this case is a parallelepiped. In two dimensional applications, the view volume reduces to the view window.

3.138 view window: A rectangle in the view plane. Projectors passing through the corners of the view window define the left, right, bottom, and top surfaces of the view volume.

3.139 viewing transformation: A transformation that maps positions in NDC coordinates to positions in NPC coordinates.

3.140 viewport: An application program specified part of normalized device coordinate space. In GKS-3D, this definition is restricted to a rectangular parallelepiped region of normalized device coordinate space used in the definition of the normalization transformation.

3.141 window: A predefined part of a virtual space. In GKS-3D, this definition is restricted to a rectangular parallelepiped region of the world coordinate space used for the definition of the normalization transformation.

3.142 workstation: GKS-3D is based on the concept of abstract graphical workstations, which provide the logical interface through which the application program controls physical devices.

3.143 workstation dependent segment storage (WDSS): Segment storage on a workstation that is used for graphical output. Segments cannot be transferred from WDSS to another workstation.

3.144 workstation independent segment storage (WISS): A special workstation type, where segments can be stored and later transferred to other workstations.

Definitions

3.145 workstation mandatory: Workstation mandatory describes a property that is required to be realized identically on all workstations of a GKS-3D implementation.

3.146 workstation transformation: A transformation which maps the contents of the workstation window into the workstation viewport, preserving aspect ratio in X and Y, but not necessarily in Z.

3.147 workstation viewport: A rectangular parallelepiped in device coordinate space. All graphics output appears within this volume.

3.148 workstation window: A rectangular parallelepiped in normalized projection coordinate space. Its contents are mapped by the workstation transformation into the workstation viewport.

3.149 world coordinate (WC): A device independent Cartesian coordinate system used by the application program for specifying graphical input and output.

GKS-3D

Introduction to GKS-3D

4 The Graphical Kernel System for Three Dimensions

4.2 Introduction to GKS-3D

The Graphical Kernel System for Three Dimensions (GKS-3D) provides a functional interface between an application program and a configuration of graphical input and output devices. The functional interface contains all basic functions for interactive and non-interactive graphics on a wide variety of graphics equipment.

4.1.1 Specification

The set of functions known as the Graphical Kernel System for Three Dimensions shall be as described in clauses 4, 5 and 6 and annexes A and B. These functions are organized in nine upward compatible levels with increasing capabilities as described in 4.10. An implementation of GKS-3D shall implement precisely the functions of one level. A GKS-3D implementation shall be invalid if it lies between two defined levels or outside the defined levels. In an implementation, all graphical capabilities that can be addressed by GKS-3D functions shall be used only via GKS-3D.

4.1.2 Registration¹⁾

For certain parameters of the functions, GKS-3D defines value ranges as being reserved for registration. The meanings of these values will be defined using the established procedures. These procedures do not apply to values and value ranges defined as being workstation or implementation dependent; these values and ranges are not standardized.

The interface is at such a level of abstraction that hardware peculiarities are shielded from the application program. As a result a simplified interface presenting uniform output primitives (POLYLINE, POLYMARKER, TEXT, FILL AREA, CELL ARRAY, GENERALIZED DRAWING PRIMITIVE), and uniform input classes (LOCATOR, STROKE, VALUATOR, CHOICE, PICK, STRING) is obtained.

In 4.3 the concepts of basic output, input and the organization of input and output sequences are outlined. A central concept both for structuring GKS-3D and for realizing device independence is introduced, called the workstation.

The facilities for picture manipulation and change are introduced via the segment facilities, the dynamic attributes and the transformations. The integral control over all these methods for change is further explained in 4.5.3 on workstations.

The concept of multiple workstations allows simultaneous output to and input from various display systems. Facilities for internal and external storage are provided by special workstations together with the possibility of transferring graphical entities directly from the special workstation for internal storage to other workstations.

Not every GKS-3D implementation needs to support the full set of functions. Nine levels are defined to meet the different requirements of graphics systems. Each GKS-3D implementation precisely provides the functions of one level. The levels are upward compatible.

GKS-3D defines only a language independent nucleus of a graphics system. For integration into a language, GKS-3D is embedded in a language dependent layer containing the language conventions, for example, parameter and name assignment.

¹⁾ For the purpose of this International Standard and according to the rules for the designation and operation of registration authorities in the ISO directives, the ISO Council has designated the National Bureau of Standards (Institute of Computer Sciences and Technology) A-266, Technology Building, Gaithersburg, MD 20899, USA to act as the registration authority.

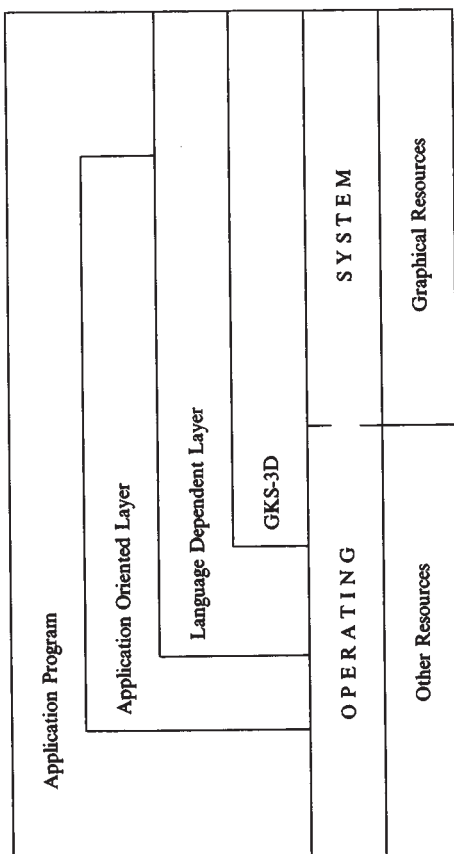


Figure 1 - Layer model of GKS-3D

The layer model represented in figure 1 illustrates the role of GKS-3D in a graphics system. Each layer may call the functions of the adjoining lower layers. In general the application program uses the application oriented layer, the language dependent layer, other application dependent layers, and operating system resources. All workstation capabilities that can be addressed by GKS-3D functions are used only via GKS-3D.

4.3 Concepts

The graphical output that is generated by GKS-3D is built up from two groups of basic elements called output primitives and primitive attributes. The output primitives are abstractions of basic actions a device can perform, such as drawing lines, and printing character strings. The attributes control the aspects of the output primitives on a device, such as linestyle, colour, character height and pick identifier. Non-geometric aspects, such as colour but not character height, can be controlled for each workstation individually, to make best use of its capabilities.

The graphical information that is input from a device, as a result of operator actions, is mapped by GKS-3D onto six classes of input each represented by a data type referred to as a logical input value. An instance of such a device representation is called a logical input device. The effect of input actions on the display space, such as prompts and echoes, is controlled by GKS-3D for each logical input device individually.

The two abstract concepts (abstract output and abstract input) are the building blocks of a so-called abstract workstation. A workstation of GKS-3D represents a unit consisting of zero or one display spaces and zero or more input devices, such as keyboard, tablet and lightpen. The workstation presents these devices to the application program as a configuration of abstract devices thereby shielding the hardware peculiarities. Conceptually, all workstations are three dimensional in GKS-3D.

The geometrical information (coordinates) contained in output primitives, attributes and logical input values (locators and strokes) can be subjected to transformations. These transformations perform mappings between five coordinate systems, namely:

- World Coordinates (WC), used by the application programmer;
- Normalized Device Coordinates (NDC), used to define a uniform coordinate system for all workstations;
- View Reference Coordinates (VRC), used to define the parameters of a view;
- Normalized Projection Coordinates (NPC), used to facilitate hidden line/hidden surface computations and composition of different views;
- Device Coordinates (DC), one coordinate system per workstation, representing its display space coordinates.

Output primitives and attributes are mapped from WC to NDC by normalization transformations, from NDC to NPC by segment transformations (see next paragraph), from NPC to VRC by view orientation transformations, from VRC to NPC by view mapping transformations, and from NPC to DC by workstation transformations. Locator and stroke input is mapped by the inverse of these transformations from DC to WC.

Output primitives and primitive attributes may be grouped together in a segment. Segments are the units for manipulation and change. Manipulation includes creation, deletion, and renaming. Change includes transforming a segment, making a segment visible or invisible, and highlighting a segment. Segments also form the basis for workstation independent storage of pictures at run time. Via this storage, which is set up as a special workstation called workstation independent segment storage, segments can be inserted and transferred to other workstations.

The attributes which control the appearance of parts of the picture (output primitives, segments, prompt and echo types of input devices) on the display space are organized in a uniform manner. Two groups of attributes apply to the appearance of each output primitive: primitive attributes (that are workstation independent) and workstation attributes. Primitive attributes are specified modally and are bound to a primitive when it is created. The primitive attributes include all geometric

aspects of primitives, such as character height for text and pattern size for fill area. In addition, the non-geometric aspects of primitives are controlled by the primitive attributes in one of two ways. A non-geometric aspect of a primitive can be specified either by an attribute that specifies one or more aspects of the primitive by an index that points to a workstation dependent representation (set of values) or by an attribute that specifies each aspect independent of the workstation. The former is referred to as bundled specification and the latter is referred to as individual specification. In addition, an index is bound to the primitive which selects the view and projection transformations in a workstation dependent manner.

Hidden lines or hidden surfaces may be removed using a workstation dependent method (see 4.5.7).

Workstation attributes include the actual representations on a workstation pointed to by indices used in bundled specification of non-geometric aspects. For example, the representations (or bundles) for polyline each contain values of linetype, linewidth scale factor and colour index. Workstation attributes also specify the colour and pattern tables and the control over deferral of picture change. Workstation attributes can be reset dynamically.

The appearance of segments is controlled by segment attributes, which are segment transformation, visibility, highlighting, segment priority, and detectability. These may be reset dynamically. Segment attributes can be a basis for feedback during manipulations (for example, highlighting).

The attributes which control the operation of logical input devices can be specified either upon initialization or as part of input device setting, depending upon the attributes. Through initialization, an initial value, a prompt and echo technique, and a volume within the display space for echoing can be specified. A data record may further provide device specific attributes. Through input device setting, the operating mode may be selected and the echo may be switched on or off. The operating modes of logical input devices specify who (operator or application program) has the initiative: SAMPLE input is acquired directly by the application program; REQUEST input is produced by the operator in direct response to the application program; EVENT input is generated asynchronously by the operator and is collected in a queue for the application program.

At run time GKS-3D can be in one of five different operating states. Associated with each state are the set of GKS-3D functions allowed in this state, and a set of state variables. The operating state concept and the state variables allow for proper specification of initializations (for example, at OPEN WORKSTATION) and the effect of various functions, especially with respect to the maintenance of device independence. One special set of functions called inquiry functions is allowed in all states. They give read-only access to the state lists. In this way useful information can be provided when errors occur. Other inquiry functions allow read-only access to the workstation descriptions, to allow the application program to adapt to particular workstation capabilities. Inquiry functions never cause errors. Instead they return information specifying whether a valid inquiry was made.

GKS-3D provides an interface to a system for filing graphical information for the purpose of external long term storage and exchange. The interface consists of a GKS-3D Metafile output workstation, which writes to a so-called graphics metafile (which is sequential), and a GKS-3D Metafile input workstation, which reads from the metafile. In addition to the normal functions for output to workstations, a GKS-3D Metafile output workstation may accept items containing non-graphical information. Input from a metafile is controlled by read and interpret functions which have the same effect as invoking the corresponding functions directly from the application program.

GKS-3D provides graphic functions for primitives defined in a three dimensional space whereas ISO 7942 provides functions for primitives defined in a plane. Thus, ISO 7942 functions are a special

GKS-3D

case of the corresponding functions in GKS-3D. The functions common to ISO 7942 and GKS-3D are called the 2D functions.

GKS-3D has a set of output primitives (see 4.4.1) which can be generated either by a 3D function or by a 2D function (e.g., a polyline primitive is generated by the 3D function POLYLINE 3 or the 2D function POLYLINE). However, 2D functions used in a 3D environment can be affected by 3D functions, for example, primitives generated using 2D functions will be transformed by 3D transformation matrices.

The primitives generated by the 2D functions are considered to lie in the $Z=0$ plane. The default view is an orthographic projection onto the $Z=0$ plane. The implication is that application programs run on a GKS-3D implementation produce the same results as on an ISO 7942 implementation.

The mapping between the 2D functions and the corresponding 3D functions is described in 4.14.

Graphical output

GKS-3D

4.4 Graphical output

4.4.1 Output primitives

The graphical information that is generated by GKS-3D and routed to all active workstations is built up of basic pieces called output primitives. GKS-3D provides seven output primitives:

- a) **POLYLINE:** GKS-3D generates a set of connected lines defined by a point sequence.
- b) **POLYMARKER:** GKS-3D generates symbols of one type centred at given positions.
- c) **TEXT:** GKS-3D generates a character string at a given position.
- d) **FILL AREA:** GKS-3D generates a polygonal area which may be hollow or filled with a uniform colour, a pattern, or a hatch style.
- e) **FILL AREA SET:** GKS-3D generates a set of polygonal areas which may be hollow or filled with a uniform colour, a pattern, or a hatch style. This allows for specifying areas with holes or disjoint regions that must be treated as a single entity, particularly for hidden line/hidden surface removal computation.
- f) **CELL ARRAY:** GKS-3D generates an array of pixels with individual colours.
- g) **GENERALIZED DRAWING PRIMITIVE(GDP):** GKS-3D addresses special geometrical output capabilities of a workstation such as the drawing of spline curves, circular arcs, and elliptic arcs. The objects are characterized by an identifier, a set of points and additional data. GKS-3D applies all transformations to the points but leaves the interpretation to the workstation.

Text, fill area, fill area set, and cell array are planar primitives in arbitrary planes. Conceptually these planes have 2 sides and zero thickness. The viewing transformation determines which side of the primitive is seen.

The plane in which the text primitive lies is defined by a point and two vectors that are contained in the definition of the text primitive. Similarly, the plane in which the cell array primitive lies is defined by three points contained in the definition of the cell array primitive. For fill area and fill area set primitives, the plane of the primitives is defined by the points of the respective primitives.

Ensuring that the points are coplanar is the responsibility of the application program. Implementations are not required to check coplanarity. While slightly non-coplanar points (e.g., owing to numerical roundoff errors) should be mapped to a plane, the effect of large deviations is implementation dependent.

GKS-3D

Graphical output

4.4.2 Output primitive attributes

Each output primitive potentially has four types of attribute (geometric, non-geometric, viewing and identification). The first three attribute types determine the exact appearance of the output primitive while the fourth attribute type is used in connection with input. The values of these attributes are set modally and are recorded in the GKS-3D state list. A separate function is provided for each primitive attribute (except the ASFs; see later in this sub-clause), to allow the application program to specify the value of an attribute without unnecessarily specifying the values of other attributes. During creation of an output primitive (that is, when one of the GKS-3D output primitive functions is invoked) these values are bound to the primitive and cannot be changed afterwards.

Attributes of the first type control the geometric aspects of primitives; these are aspects which affect the shape or size of the whole primitive (for example, CHARACTER HEIGHT for TEXT). Hence, they are sometimes referred to as geometric attributes. Attributes of this type are workstation independent and, if they represent coordinate data (points or displacements), are expressed in world coordinates (for example, CHARACTER HEIGHT is expressed in world coordinates but TEXT PATH takes one of a set of enumerated values). They are defined separately for each primitive and a primitive may have zero, one or many geometric attributes.

Current values of (workstation independent) geometric attributes, which are expressed in world coordinates, are stored in world coordinates. When they are bound to their respective primitives, the values are subject to the same transformations as the geometric data contained in the definition of the primitive. Hence, current values are unaffected by changes in the normalization, viewing and workstation transformation.

Attributes of the second type control the non-geometric aspects of primitives; these are aspects which merely affect a primitive's appearance (for example, linetype for POLYLINE, or colour index for all primitives except CELL ARRAY) or the shape or size of the component parts of the primitive (for example, marker size scale factor for POLYMARKER). Non-geometric aspects do not represent coordinate data. The non-geometric aspects of a primitive may be specified in one of two ways, namely via a bundle or individually.

Bundled aspects are controlled by a bundle index into a bundle table, each entry of which contains non-geometric aspects of a primitive. The non-geometric aspects are workstation dependent in this method of specification and each workstation has its own set of bundle tables (stored in the workstation state list). The values in a particular bundle (or entry in the bundle table) may be different for different workstations.

For individual specification of aspects, there is a separate attribute for each non-geometric aspect. As with the attributes controlling the geometric aspects, these attributes are workstation independent and are stored in the GKS-3D state list.

For a given non-geometric aspect, the values that can be assigned to the appropriate bundle component are the same as the values that can be assigned to the corresponding attribute for individual specification. Since the bundles are set separately for each workstation, the values of their components are restricted to the valid values for that workstation. In the case of individual attribute specification, such restrictions are not imposed. Default actions for the display of a primitive are defined to occur if it is created with a value of an individually specified attribute that is invalid on a particular workstation.

As indicated above, GENERALIZED DRAWING PRIMITIVE (GDP) and CELL ARRAY do not have associated bundle tables nor corresponding individually specified attributes. The GDP may use the most appropriate bundle tables or sets of individually specified attributes for each GDP

Graphical output

GKS-3D

function. For example, if one GDP function is essentially a FILL AREA, then the fill area bundle table or the set of individually specified fill area attributes would be used. CELL ARRAY contains colour index information as part of its definition but has no other non-geometric aspects and so does not use a bundle table nor does it have a set of individually specified attributes.

The method of specification of the non-geometric aspects of a primitive may be chosen separately for each aspect. A further group of primitive attributes, called ASPECT SOURCE FLAGS (ASFs), take the values INDIVIDUAL and BUNDLED to specify the choice. As with the other primitive attributes, these attributes are workstation independent and are stored in the GKS-3D state list. There is one ASF for each non-geometric aspect. The initial values of all the ASFs are the same; this may be either BUNDLED or INDIVIDUAL, the choice being implementation dependent. If the initial values are not altered, the system will operate

- a) as if individual specification of non-geometric aspects were not a system feature, if the initial values of all the ASFs are BUNDLED;
- b) as if specification of non-geometric aspects via a bundle were not a system feature, if the initial values of all the ASFs are INDIVIDUAL.

The flags may be set at any time when GKS-3D is open by the functions SET ASPECT SOURCE FLAGS and SET ASPECT SOURCE FLAGS 3. This enables some non-geometric aspects of a primitive to be specified individually and others via a bundle.

When a primitive is displayed, the values of the non-geometric aspects with which it is displayed are determined as follows.

- c) If the ASF for an aspect is INDIVIDUAL, the value used on all workstations is the value of the corresponding individually specified attribute of that primitive.
- d) If the ASF for an aspect is BUNDLED, the value used on a workstation is obtained via the bundle table for that primitive on the workstation; the corresponding component of the bundle, pointed to by the bundle index, is used.

If colour is a non-geometric aspect of a primitive, then it is specified as an index into a separate colour table. There is only one colour table per workstation into which all the colour indices point. Similarly, other entries in a bundle, or corresponding individually specified attributes, may be indices either into another workstation table (for example, style index when interior style PATTERN is used) or into a fixed list (for example, linetypes for polyline).

There are two attributes of the third type, namely VIEW INDEX and HLHSR IDENTIFIER. The attribute VIEW INDEX is a pointer to a view table entry at a workstation. There is one view table per workstation.

The attribute HLHSR IDENTIFIER supplies HLHSR information to the workstation. The interpretation of the identifier is implementation dependent (see 4.5.7). Implementation of the functions to control hidden line/hidden surface removal is obligatory, but the functions may have no effect on the appearance of primitives.

There is precisely one attribute of the fourth type per primitive, namely PICK IDENTIFIER. This is used for identifying a primitive, or a group of primitives, in a segment when that segment is picked.

The attributes which apply to each output primitive (attributes controlling non-geometric aspects, geometric attributes, view attributes and PICK IDENTIFIER) are:

CKS-3D

e) POLYLINE:

POLYLINE INDEX
LINETYPE
LINEWIDTH SCALE FACTOR
POLYLINE COLOUR INDEX
LINETYPE ASF
LINEWIDTH SCALE FACTOR ASF
POLYLINE COLOUR INDEX ASF
VIEW INDEX
HLHSR IDENTIFIER
PICK IDENTIFIER

f) POLYMARKER:

POLYMARKER INDEX
MARKER TYPE
MARKER SIZE SCALE FACTOR
POLYMARKER COLOUR INDEX
MARKER TYPE ASF
MARKER SIZE FACTOR ASF
POLYMARKER COLOUR INDEX ASF
VIEW INDEX
HLHSR IDENTIFIER
PICK IDENTIFIER

g) TEXT:

TEXT INDEX
TEXT FONT AND PRECISION
CHARACTER EXPANSION FACTOR
CHARACTER SPACING
TEXT COLOUR INDEX
TEXT FONT AND PRECISION ASF
CHARACTER EXPANSION FACTOR ASF
CHARACTER SPACING ASF
TEXT COLOUR INDEX ASF
CHARACTER HEIGHT
CHARACTER UP VECTOR
TEXT PATH
TEXT ALIGNMENT
VIEW INDEX
HLHSR IDENTIFIER
PICK IDENTIFIER

h) FILL AREA:

FILL AREA INDEX
FILL AREA INTERIOR STYLE
FILL AREA STYLE INDEX
FILL AREA COLOUR INDEX
FILL AREA INTERIOR STYLE ASF
FILL AREA STYLE INDEX ASF
FILL AREA COLOUR INDEX ASF
PATTERN SIZE
PATTERN REFERENCE POINT AND VECTORS
VIEW INDEX
HLHSR IDENTIFIER
PICK IDENTIFIER

- i) FILL AREA SET:

FILL AREA INDEX
FILL AREA INTERIOR STYLE
FILL AREA COLOUR INDEX
FILL AREA INTERIOR STYLE ASF
FILL AREA STYLE INDEX ASF
FILL AREA COLOUR INDEX ASF
PATTERN SIZE
PATTERN REFERENCE POINT AND VECTORS
EDGE INDEX
EDGE FLAG
EDGETYPE
EDGEWIDTH SCALE FACTOR
EDGE COLOUR INDEX
EDGE FLAG ASF
EDGEWIDTH SCALE FACTOR ASF
EDGE COLOUR INDEX ASF
VIEW INDEX
HLHSR IDENTIFIER
PICK IDENTIFIER

- j) CELL ARRAY:

PICK IDENTIFIER
VIEW INDEX
HLHSR IDENTIFIER

- k) GENERALIZED DRAWING PRIMITIVE
- Zero or more of the sets e) to i) except that PICK IDENTIFIER, VIEW INDEX and HLHSR IDENTIFIER are always attributes

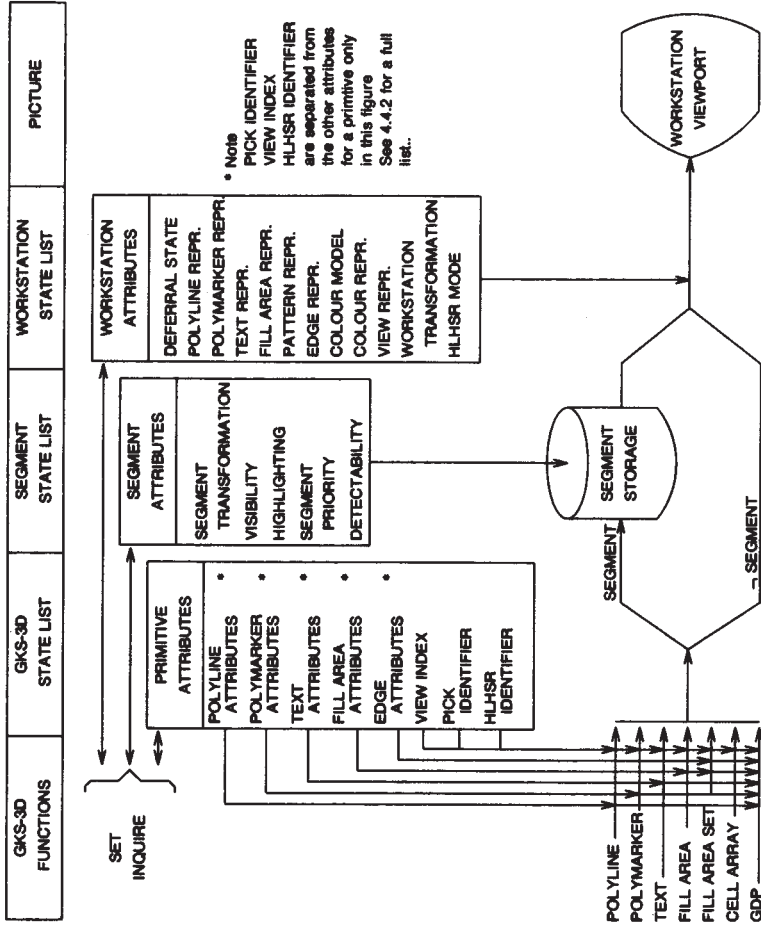


Figure 2 - Binding of attributes

Figure 2 shows the binding of the attributes.

The attributes for each primitive, other than PICK IDENTIFIER, are described in 4.4.3 to 4.4.13. PICK IDENTIFIER is described in more detail in 4.7.1. VIEW INDEX and HLHSR IDENTIFIER are described in more detail in 4.4.12 and 4.4.13. In the descriptions, attributes appear in upper case (for example, the attributes CHARACTER HEIGHT and PICK IDENTIFIER); aspects appear in both upper and lower case, according to their context. Geometric aspects are always controlled by geometric attributes and so appear in upper case (for example, the aspect CHARACTER HEIGHT). Non-geometric aspects may be controlled via a <bundle> INDEX or by individually specified attributes. Non-geometric aspects appear in lower case unless the corresponding individually specified attributes are being used which appear in upper case (for example, the aspect linetype but the individually specified attribute LINETYPE).

The entries in the bundle, pattern, colour, and view tables may be set separately for each workstation. Some standard definitions for table entries are contained in the workstation description table and are used as initial values. The application program may select a standard definition or may define the values of a specific entry explicitly. Only the most commonly used (or anticipated) combinations of values need to be predefined for each output type workstation. At least those predefined entries with indices up to the minimum number of predefined entries at a given level (see 4.10.3) are distinguishable from each other. Other combinations of values can be specified by the SET <bundle | PATTERN | COLOUR> REPRESENTATION and SET VIEW REPRESENTATION 3 functions, possibly after inquiring the workstation capabilities. The tables, which are on every workstation of category OUTPUT, OUTIN or MO (i.e. they are workstation attributes), are:

polyline bundle table
 polymarker bundle table
 text bundle table
 fill area bundle table
 edge bundle table
 pattern table
 colour table
 view table

The values in these tables may be (dynamically) changed. In fact, the only way of changing the aspects of a primitive which are stored in a bundle table is by changing that table. However, note that a change in a bundle table entry can only be reflected in a displayed primitive if the values of the corresponding ASFs (of that primitive) for the aspects in the bundle table are BUNDLED. The entry 'dynamic modification accepted' in the workstation description table indicates which changes

l) lead to an implicit regeneration (may be deferred) (IRG);

m) can be performed immediately (IMM).

The deferral state is explained in more detail in 4.5.3. If changes can be performed immediately, those changes may affect primitives outside segments in addition to those inside segments.

4.4.3 Polyline attributes

Polyline has no geometric attributes. The representation of polyline on the workstation is controlled by the POLYLINE INDEX, or the set of individually specified polyline attributes (LINETYPE, LINEWIDTH SCALE FACTOR, and POLYLINE COLOUR INDEX) or some combination of the two, depending upon the values of the ASFs for linetype, linewidth scale factor and polyline colour index. The POLYLINE INDEX is a pointer into the polyline bundle table, each entry of which contains values for linetype, linewidth scale factor and polyline colour index.

Linetypes 1 to 4 are solid, dashed, dotted and dashed-dotted. Every workstation of category

OUTPUT or OUTIN realizes linetypes 1 to 4 with recognizable styles. Linetypes greater than 4 are reserved for registration (see 4.1.2). Linetypes less than 0 may be available but their styles are implementation dependent. The linetype specifies a sequence of line segments and gaps which are repeated to draw a polyline. It is workstation dependent whether this sequence is restarted or continued at the start of the polyline, at the start of a clipped piece of a polyline, and at each vertex of a polyline.

The linewidth is calculated as a nominal linewidth multiplied by the linewidth scale factor. This value is mapped by the workstation to the nearest available linewidth.

4.4.4 Polymarker attributes

Polymarker has no geometric attributes. The representation of polymarker at the workstation is controlled by the POLYMARKER INDEX, or the set of individually specified polymarker attributes (MARKER TYPE, MARKER SIZE SCALE FACTOR, and POLYMARKER COLOUR INDEX) or some combination of the two, depending upon the values of the ASFs for marker type, marker size scale factor, and polymarker colour index. The POLYMARKER INDEX is a pointer into the polymarker bundle table, each entry of which contains values for marker type, marker size scale factor and polymarker colour index.

Marker types 1 to 5 are dot, plus sign, asterisk, circle, and diagonal cross each centred on the positions they are identifying. Every workstation of category OUTPUT or OUTIN realizes marker types 1 to 5 with recognizable shapes at the given positions. Marker types greater than 5 are reserved for registration (see 4.1.2). Marker types less than 0 may be available but their forms are implementation dependent.

The marker size is calculated as a nominal size multiplied by the marker size scale factor. This size is mapped by the workstation to the nearest available size. Marker type 1 is always displayed as the smallest displayable dot.

The marker is visible if, and only if, the marker position is within the clipping volume. The clipping of partially visible markers is workstation dependent. Since the polymarker has no geometric attributes, transformations affect only the marker position and not the appearance of the marker glyph.

4.4.5 Text attributes

Text is a planar primitive. The plane of the text primitive is defined by a point and two vectors contained in the definition of the text primitive. The point is called the text position. The two vectors are called the first and second text direction vectors. The point and these vectors define a text plane and a local coordinate system on that plane in which all geometric text attributes are measured. This coordinate system, which is known as the text local coordinate (TLC) system, will be defined in the next paragraph.

Let P be the text position and U and V be the first and second text direction vectors respectively. The text plane is the plane that contains P and is perpendicular to $U \times V$. The origin of the local coordinate system is at the point P. The positive X axis of the local coordinate system is parallel to U. Let T be a vector perpendicular to U and in the text plane. If T.V is positive, then the positive Y axis is parallel to T otherwise the positive Y axis is opposite T. By this definition, any point in the text plane can be represented by two reals r_1 and r_2 , since it is uniquely defined by the expression $P + r_1 E_1 + r_2 E_2$, where E_1 and E_2 are two unit vectors parallel to the positive X axis and Y axis, respectively. This means that there is a 1-1 correspondence between a pair of two reals and a point in the text plane. In a similar way, there is a 1-1 correspondence between a pair of two reals and a vector parallel to the text plane. This correspondence holds especially for the 2D vectors

CHARACTER UP VECTOR and the CHARACTER BASE VECTOR, which will be later be defined in this subclause.

In the following all references to coordinates refer to this text local coordinate system unless explicitly stated otherwise.

Text has the geometric attributes CHARACTER HEIGHT, CHARACTER UP VECTOR, TEXT PATH, and TEXT ALIGNMENT which are specified and used as described in this sub-clause.

Text also has two implicitly specified geometric attributes CHARACTER WIDTH and CHARACTER BASE VECTOR. These are implicitly specified by the functions SET CHARACTER HEIGHT and SET CHARACTER UP VECTOR respectively. They otherwise behave like ordinary geometric attributes (their values are bound to TEXT primitives when the primitives are created and cannot be changed afterwards and these values are subject to the same transformations as the geometric data contained in the definition of the primitive).

The representation of text at the workstation is controlled by the TEXT INDEX, or the set of individually specified text attributes (TEXT FONT AND PRECISION, CHARACTER EXPANSION FACTOR, CHARACTER SPACING, and TEXT COLOUR INDEX) or some combination of the two, depending upon the values of the ASFs for text font and precision, character expansion factor, character spacing and text colour index. The TEXT INDEX is a pointer into the text bundle table, each entry of which contains values for text font and precision, character expansion factor, character spacing and text colour index.

Precise control of the appearance of TEXT on a workstation is provided by the following aspects: CHARACTER HEIGHT, CHARACTER WIDTH, character expansion factor, TEXT PATH, CHARACTER UP VECTOR, CHARACTER BASE VECTOR, character spacing and TEXT ALIGNMENT. However, the use of these values in displaying text is determined by the setting of the text font and precision aspect (font and precision are two components of the same aspect). The CHARACTER HEIGHT specifies the nominal height of a capital letter character. The CHARACTER WIDTH specifies the nominal width of a character; the actual width depends upon the width to height ratio of the character indicated by the font designer and may vary from character to character. The character expansion factor specifies the deviation of the width to height ratio of the character from the ratio indicated by the font designer. The CHARACTER UP VECTOR gives the up direction of a character. The CHARACTER BASE VECTOR gives the direction of the baseline of a character. Only the directions, not the lengths, of these vectors are relevant. TEXT PATH has the possible values RIGHT, LEFT, UP and DOWN. It specifies the writing direction of the text string. For RIGHT, the text string is written along a baseline in the direction of the CHARACTER BASE VECTOR. For LEFT, the baseline direction is the opposite direction of the CHARACTER BASE VECTOR. For UP, the character path coincides with the direction of the CHARACTER UP VECTOR. For DOWN, it is the opposite direction of the CHARACTER UP VECTOR. For the UP and DOWN text path directions the characters are arranged so that the centres of the character bodies are on a straight line in the direction of the CHARACTER UP VECTOR.

The character spacing value specifies how much additional space is to be inserted between two adjacent character bodies. If the value of character spacing is zero, the character bodies are arranged one after the other along the TEXT PATH, without any additional space between. A positive value of character spacing will insert additional space between character bodies. A negative value of character spacing will cause adjacent character bodies to overlap. Character spacing is specified as a fraction of the font nominal character height.

The effect of the aspects CHARACTER HEIGHT, CHARACTER WIDTH, character expansion factor, TEXT PATH, character spacing and text font is to define an (imaginary) rectangle with its

sides parallel to the x and y axes, enclosing the text. The bounds of this enclosing rectangle are as follows. For TEXT PATH = LEFT or RIGHT, the height of the rectangle is the height of the character body of the specified font; the left side of the rectangle is the left side of the character body of the leftmost character and the right side of the rectangle is the right side of the character body of the rightmost character. For TEXT PATH = UP or DOWN, the top of the rectangle is the top of the character body of the topmost character and similarly, the bottom of the rectangle is the bottom of the bottommost character; the width of the rectangle is the width of the widest character in the specified font.

The effect of the CHARACTER UP VECTOR and CHARACTER BASE VECTOR attributes is to transform the enclosing rectangle, thus defining an enclosing parallelogram, the text extent parallelogram (the rectangle has been rotated and sheared).

The TEXT ALIGNMENT attribute controls the positioning of this text extent parallelogram in relation to the text position. For simplicity the TEXT ALIGNMENT is described in terms of the default CHARACTER UP VECTOR and CHARACTER BASE VECTOR, when the text extent parallelogram is actually a rectangle. The horizontal component of TEXT ALIGNMENT has four values: LEFT, CENTRE, RIGHT and NORMAL. If the horizontal component is LEFT, the left side of the text extent parallelogram passes through the text position. Similarly, if the value is RIGHT, the right side of the text extent parallelogram passes through the text position. If the horizontal component is CENTRE, the text position lies midway between the left and right sides of the text extent parallelogram. Thus, if TEXT PATH = UP or DOWN, the straight line passing through the centrelines of the characters also passes through the text position. The vertical component of TEXT ALIGNMENT has six values: TOP, CAP, HALF, BASE, BOTTOM and NORMAL. These each correspond to one of the font specific horizontal lines in the definition of a character (see figure 3). A value of TOP causes the top of the text extent parallelogram to pass through the text position. A value of CAP causes the text position to lie on the capline of the whole string (TEXT PATH = LEFT or RIGHT) or on the capline of the topmost character in the string (TEXT PATH = UP or DOWN). A value of HALF causes the text position to lie on the halfline of the whole string (TEXT PATH = LEFT or RIGHT) or on a line halfway between the halflines of the top and bottom characters (TEXT PATH = UP or DOWN). A value of BASE causes the text position to lie on the baseline of the whole string (TEXT PATH = LEFT or RIGHT) or on the baseline of the bottom character in the string (TEXT PATH = UP or DOWN). A value of BOTTOM causes the bottom of the text extent parallelogram to pass through the text position.

In the general case, the orientation referred to as horizontal is that of the CHARACTER BASE VECTOR with RIGHT representing direction of that vector and LEFT being opposite to it. Similarly the orientation referred to as vertical is that of the CHARACTER UP VECTOR with UP representing the direction of that vector and DOWN being opposite to it.

Either component of TEXT ALIGNMENT can take the value NORMAL. For each value of TEXT PATH, the effect of a particular component being NORMAL is equivalent to one of the other values of that component. In each case, the equivalent alignment value is chosen to achieve a natural alignment for that TEXT PATH value. The complete list of equivalent values is:

TEXT PATH	NORMAL Horizontal and Vertical Alignments
RIGHT	(LEFT, BASE)
LEFT	(RIGHT, BASE)
UP	(CENTRE, BASE)
DOWN	(CENTRE, TOP)

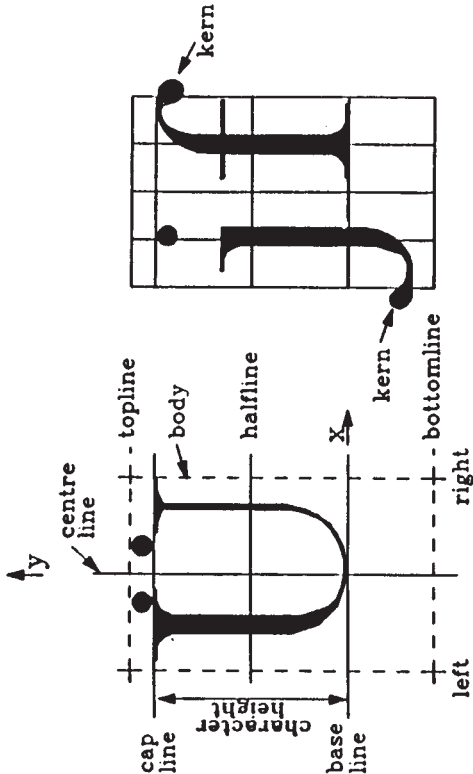
a) STRING: The TEXT character string is generated in the requested text font and is positioned by aligning the TEXT output primitive at the given text position. CHARACTER HEIGHT, CHARACTER WIDTH and character expansion factor are evaluated as closely as reasonable, given the capabilities of the workstation. CHARACTER UP VECTOR, CHARACTER BASE VECTOR, TEXT PATH, TEXT ALIGNMENT and character spacing, need not be used. Clipping is done in an implementation and workstation dependent way.

b) CHAR: The TEXT character string is generated in the requested text font. For the representation of each individual character, the aspects CHARACTER UP VECTOR, CHARACTER WIDTH, the up direction of the CHARACTER UP VECTOR, the baseline direction of the CHARACTER BASE VECTOR, and character expansion factor are evaluated as closely as possible, in a workstation dependent way. The spacing used between character bodies is evaluated exactly; the character body, for this purpose, is an ideal character body, calculated precisely from the text aspects and the font dimensions. The position of the resulting text extent parallelogram is determined by the TEXT ALIGNMENT and the text position. Clipping is performed at least on a character by character basis.

c) STROKE: The text character string in the requested text font is displayed at the text position by applying all text aspects. The character string is clipped exactly at the clipping volume.

STROKE precision does not necessarily mean vector strokes; as long as the representation adheres to the rules governing STROKE precision, the font may be realized in any form, for example by raster fonts.

Figure 3 - Font description coordinate system



The initial values of the geometric text attributes are:

CHARACTER HEIGHT	TLC	0.01 (i.e. 1% of the height of the default window)
CHARACTER UP VECTOR	TLC	(0.1) RIGHT
TEXT PATH		
TEXT ALIGNMENT		(NORMAL, NORMAL)

and the initial values of the implicitly specified geometric text attributes are:

CHARACTER WIDTH	TLC	0.01 (i.e. the same value as the initial value of CHARACTER HEIGHT)
CHARACTER BASE VECTOR	TLC	(1.0)

Text font and precision together constitute one aspect. The text font value is used to select a particular font on the workstation. Every workstation supports at least one font that is able to generate a graphical representation of the characters defined in ISO 646. This is font number 1. Font numbers greater than 1 are reserved for registration (see 4.1.2). Font numbers less than 0 may be supported but are implementation dependent.

The text precision value is used to select the 'closeness' of the text representation at the workstation in relation to that defined by the workstation independent text attributes and the transformation and clipping currently applicable. The text precision value has the following possible values:

Graphical output

The height of a character in the font coordinate system is given by the height from the font base line to the capline. The width of a character is given by the width of the character body. The width of a character may include space on either side of the character and this space is generally evenly split between the left and right sides of the character. It is assumed that the characters lie within their body, except that kerned characters may exceed the side limits of the character body.

In general, the top limits of the bodies for a font are identical with, or very close to, the typographical capline or ascender line, and the bottom limit to the descender line. The space, if any, between the topline and the capline may be used for an additional mark over the character, for example an accent. However, these and other details are purely for the use of the font designer. The intention is only that characters placed with their bodies touching in the horizontal direction should give an appearance of good normal spacing, and characters touching in the vertical direction will avoid clashes between ascenders and descenders (typographically 'set solid').

Since the values of CHARACTER HEIGHT, CHARACTER WIDTH, CHARACTER UP VECTOR and CHARACTER BASE VECTOR are given in text local coordinate units, but the characters are generated on the workstation in device coordinates, using the workstation dependent font and precision, the geometric attributes need to be transformed in such a way that the workstation can generate the characters in the way intended.

GKS-3D

Graphical output



CHARACTER HEIGHT = 1, CHARACTER UP VECTOR = (0,1),
TEXT PATH = RIGHT, TEXT ALIGNMENT = (NORMAL,NORMAL)



CHARACTER HEIGHT = 0.5, CHARACTER UP VECTOR = (0,1)
TEXT PATH = RIGHT, TEXT ALIGNMENT = (NORMAL,NORMAL)



CHARACTER HEIGHT = 1, CHARACTER UP VECTOR = (0,1),
TEXT PATH = RIGHT, TEXT ALIGNMENT = (RIGHT,TOP)



CHARACTER HEIGHT = 1
CHARACTER UP VECTOR = (-1,0)
TEXT PATH = RIGHT
TEXT ALIGNMENT = (NORMAL,NORMAL)

X text position
----- baseline or centreline
----- text extent rectangle
(indicated for PATH = DOWN)

NOTES

- 1 Examples are illustrated with STROKE precision, a character expansion factor of 1 and a zero character spacing.
- 2 Capline = topline in these examples.
- 3 Changed attributes are underlined.

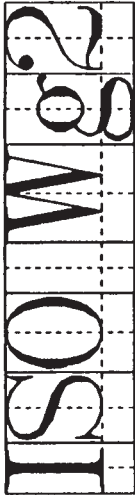
Figure 4 - Effects of changes in geometric text attributes

Graphical output

GKS-3D

GKS-3D

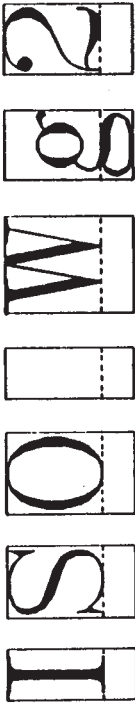
Graphical output



CHARACTER EXPANSION FACTOR = 1, CHARACTER SPACING = 0



CHARACTER EXPANSION FACTOR = 0.75, CHARACTER SPACING = 0

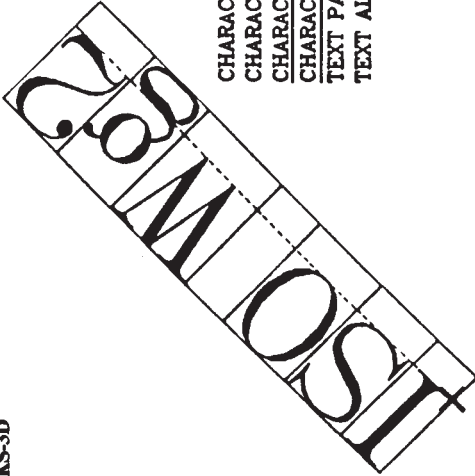


CHARACTER EXPANSION FACTOR = 1, CHARACTER SPACING = 0.3

NOTES

- 1 Examples are illustrated with default values of the geometric text attributes and with STROKE precision.
- 2 Changed aspects are underlined.

Figure 5 - Effects of changes in non-geometric text aspects



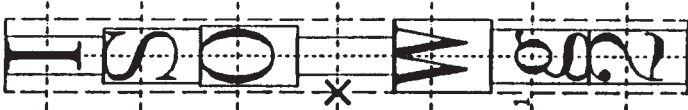
CHARACTER EXPANSION FACTOR = 1
CHARACTER SPACING = 0
CHARACTER HEIGHT = 1.414
CHARACTER UP VECTOR = (-1,1)
TEXT PATH = RIGHT
TEXT ALIGNMENT = (NORMAL,NORMAL)



CHARACTER EXPANSION FACTOR = 1
CHARACTER SPACING = -0.3
CHARACTER HEIGHT = 1
CHARACTER UP VECTOR = (0,1)
TEXT PATH = LEFT
TEXT ALIGNMENT = (NORMAL,NORMAL)



CHARACTER EXPANSION FACTOR = 1
CHARACTER SPACING = 0
CHARACTER HEIGHT = 1
CHARACTER UP VECTOR = (-1,0)
TEXT PATH = RIGHT
TEXT ALIGNMENT = (CENTRE,TOP)



CHARACTER EXPANSION FACTOR = 1
CHARACTER SPACING = 0
CHARACTER HEIGHT = 1
CHARACTER UP VECTOR = (0,1)
TEXT PATH = DOWN
TEXT ALIGNMENT = (LEFT,HALF)

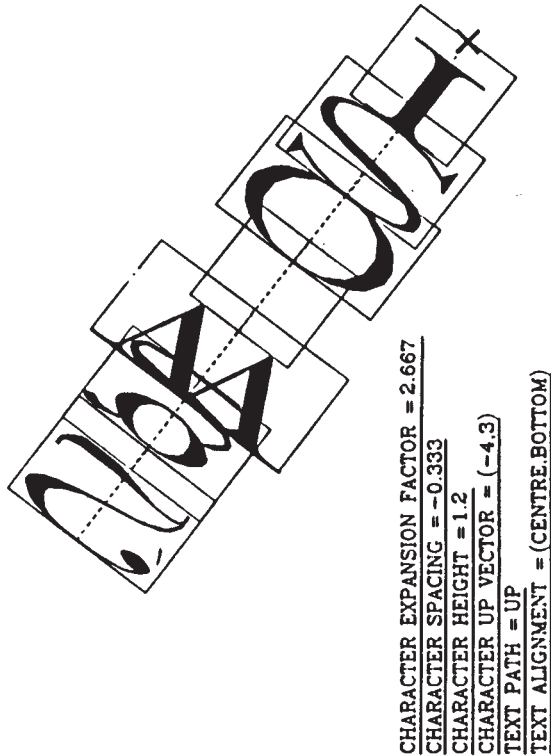
NOTES

- 1 Changes from the top example of figures 4 and 5 are underlined.
- 2 In the last example, halflines of all characters are shown.

Figure 6 - Effects of combined changes in text aspects

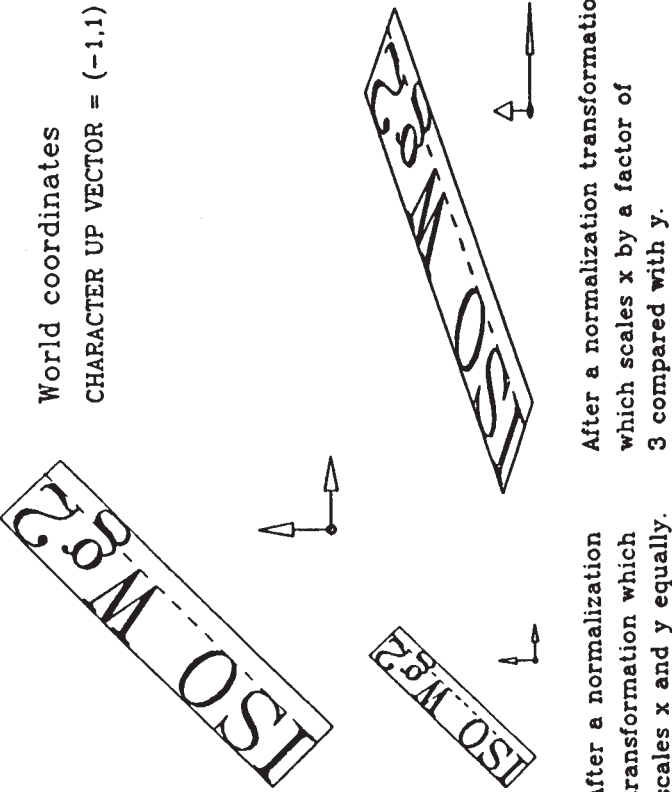
GKS-3D

GKS-3D



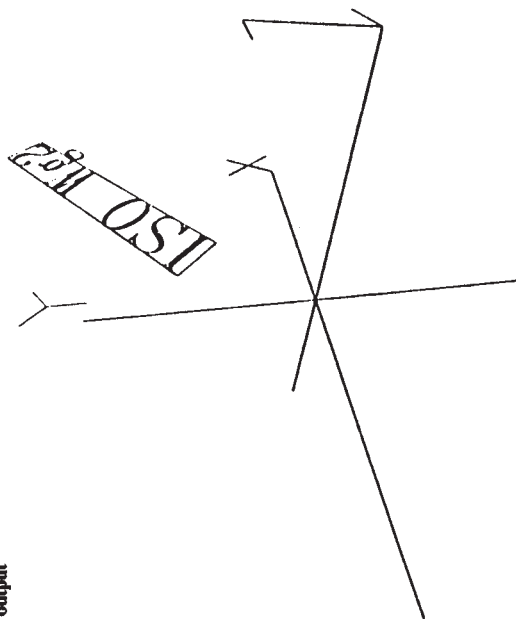
CHARACTER EXPANSION FACTOR = 2.667
CHARACTER SPACING = -0.333
CHARACTER HEIGHT = 1.2
CHARACTER UP VECTOR = (-4.3)
TEXT PATH = UP
TEXT ALIGNMENT = (CENTRE.BOTTOM)

NOTE - Changes from the top example of figures 4 and 5 are underlined.
Figure 7 - Effects of several changes in text aspects

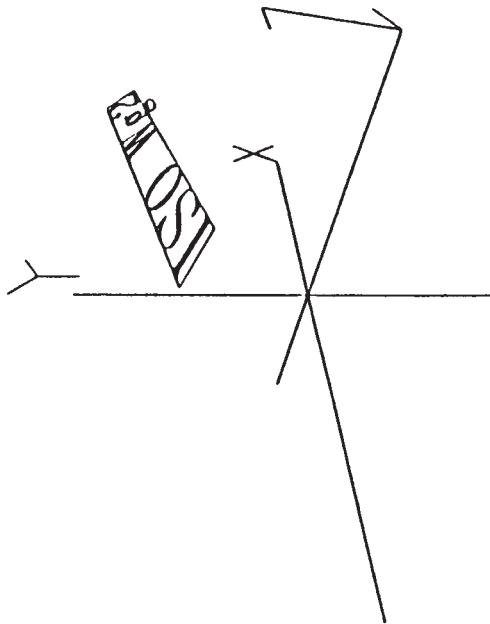


World coordinates
CHARACTER UP VECTOR = (-1.1,1)
After a normalization transformation which scales x by a factor of 3 compared with y.
After a normalization transformation which scales x and y equally.

Figure 8 - Effects of different normalization transformations on text in STROKE precision



world coordinates: character up vector = (-1,1)



After a normalization transformation which scales x and z by a factor of 2 compared with y.

Figure 9 - Effects of different normalization transformations on text in STROKE precision

The effect to be achieved is now described. A series of rectangles, aligned and positioned on the text plane by applying the current text attributes to the nominal character body extent rectangle, is passed through the viewing pipeline. The rectangle height vector has the length CHARACTER HEIGHT and is parallel to the actual character up vector (i.e. the 3D vector in WC that is equal to $ch_u E_1 + ch_u E_2$, where ch_u and ch_u are the components of the 2D CHARACTER UP VECTOR and E_1 and E_2 are unit vectors parallel to the positive X axis and Y axis in the text plane). The rectangle width vector has the length CHARACTER WIDTH and is parallel to the actual character base vector (i.e. the 3D vector in WC that is equal to $ch_b E_1 + ch_b E_2$, where ch_b and ch_b are the components of the 2D CHARACTER BASE VECTOR). The world coordinates of the four corners of the rectangle are transformed by the normalization transformation, by a segment transformation if within a segment, by a viewing transformation, and by the workstation transformation. The resulting quadrilateral can be used by the workstation character generator. Thus, the shape of individual characters can be transformed by a normalization transformation that scales x, y and z non-uniformly, by a segment transformation, by a viewing transformation and by a workstation transformation that scales z non-uniformly. Before the viewing transformation, the character body extent is a parallelogram and can be conveniently represented by the text position and two vectors to the two adjacent corners of the parallelogram. This parallelogram is stored in segments.

The resulting characters need to conform to the shape of the character extent quadrilateral exactly for STROKE precision. For STRING precision text, the character height and width vectors need to be evaluated only at the leading edge of the first character. For CHAR precision, the character height and width vectors need to be evaluated only at the leading edge of each character.

On the workstation, the height of a character is given by the length of the transformed height vector; the character up direction is given by the direction of the transformed height vector; the width of a character is given by the length of the transformed width vector multiplied by the font width to height ratio for the character and by the character expansion factor; the character base direction is given by the direction of the transformed width vector. The characters are arranged together in a text extent parallelogram, depending on the values of TEXT PATH and character spacing. The text extent parallelogram is then positioned according to the value of TEXT ALIGNMENT and the text position, contained in the definition of the TEXT primitive.

Figures 4 to 7 give examples of the effects of different values of text aspects. Figures 8 and 9 give examples of the effect of different normalization transformations on the displayed form of the text.

4.4.6 Text extent and concatenation

The text extent rectangle is the minimum rectangle which completely encloses the character bodies of the text string. This may be determined by INQUIRE TEXT EXTENT or INQUIRE TEXT EXTENT 3 which calculates the value of the text extent rectangle.

For font 1, the results are workstation independent. For other fonts, the results may be workstation dependent. STROKE precision text is assumed so that the information retrieved is exact.

4.4.7 Fill area attributes

Fill area has the geometric attributes PATTERN REFERENCE POINT and two PATTERN REFERENCE VECTORS. It also has two implicitly specified geometric attributes PATTERN WIDTH VECTOR and PATTERN HEIGHT VECTOR. These are implicitly specified by the function SET PATTERN SIZE. Like ordinary geometric attributes, their values are bound to the fill area primitives when the primitives are created and cannot be changed afterwards. These values are subject to the same transformations as the geometric data contained in the definition of the primitive. The usage of the fill area geometric attributes is described later in this sub-clause.

Graphical output

GKS-3D

The representation of fill area at the workstation is controlled by the **FILL AREA INDEX**, or the set of individually specified fill area attributes (**FILL AREA INTERIOR STYLE**, **FILL AREA STYLE INDEX**, and **FILL AREA COLOUR INDEX**) or some combination of the two, depending upon the values of the ASFs for fill area interior style, fill area style index, and fill area colour index. The **FILL AREA INDEX** is a pointer into the fill area bundle table, each entry of which contains values for the fill area interior style, fill area style index and fill area colour index.

The fill area interior style is used to determine in what style the area should be filled. It has the following values:

- a) **HOLLOW**: No filling, but draw the bounding polyline, using the fill area colour index currently selected (either via the fill area bundle or individually, depending upon the corresponding ASF). The linetype and linewidth are implementation dependent.
- b) **SOLID**: Fill the interior of the polygon using the fill area colour index currently selected (either via the fill area bundle or individually, depending upon the corresponding ASF).
- c) **PATTERN**: Fill the interior of the polygon using the fill area style index currently selected (either via the fill area bundle or individually, depending upon the corresponding ASF) as an index into the pattern table. In this context, the fill area style index is sometimes referred to as the pattern index.
- d) **HATCH**: Fill the interior of the polygon using the fill area colour index and the fill area style index currently selected (either via the fill area bundle or individually, depending upon the corresponding ASFs). The fill area style index is used as a pointer into the list of hatch styles, in which case it is sometimes referred to as the hatch index.

For interior style **PATTERN**, the pattern is defined by the pattern representation, which specifies an array ($DX \times DY$) of colour indices that are pointers into the colour table. The size and starting position of the pattern are determined by a pattern box in a local coordinate system on the fill area plane. The pattern box is defined as follows:

- e) The **PATTERN REFERENCE POINT** and the **PATTERN REFERENCE VECTORS** are projected along a normal vector onto the fill area plane.
- f) The origin of the box, which may be a parallelogram, is the projection of the **PATTERN REFERENCE POINT**.
- g) The **PATTERN WIDTH VECTOR** is one side of the pattern box. Its direction is parallel to the projection of the first **PATTERN REFERENCE VECTOR**. Its length is equal to that of the SX component of the current **PATTERN SIZE**.
- h) The **PATTERN HEIGHT VECTOR** is the adjacent side of the pattern box. Its direction is parallel to the projection of the second **PATTERN REFERENCE VECTOR**. Its length is equal to the SY component of the current **PATTERN SIZE**.

If the projected pattern box is degenerate, an implementation dependent technique determines the location and shape of the pattern box.

The pattern box is conceptually divided into a grid of $DX \times DY$ cells. The colour index array is associated with the cells as follows: the element (1,DY) is associated with the cell having the projected **PATTERN REFERENCE POINT** at one corner; elements with increasing first dimension are associated with successive cells in the direction of the **PATTERN WIDTH VECTOR**. Elements with decreasing second dimension are associated with successive cells in the direction of the **PATTERN HEIGHT VECTOR**. The pattern is mapped onto the fill area by replicating the transformed pattern box in directions parallel to its sides until the entire fill area is covered. The resulting

GKS-3D

pattern is subject to the same transformations as the fill area primitive.

Mapping the transformed pattern cells to the pixels of a raster display is performed by the following rules:

- i) If the centre of a pixel lies inside the parallelogram defined by the transformed cell, its colour is set;
 - j) If the pixel centre lies on a cell boundary, it is implementation dependent which of the adjacent cells is used to define the colour of the pixel;
 - k) The pixel is assigned the colour of the cell corresponding to the pixel's centre.
- For a workstation which can implement patterns but not transformable patterns, a suitable action is to generate non-transformed patterns to fill a polygon.

For interior style **HATCH**, the hatch index selects among hatch styles: hatch styles greater than 0 are reserved for registration (see 4.1.2); hatch styles less than 0 are workstation dependent. Whether hatching is affected by transformations or not is workstation dependent.

Interior style **HOLLOW** is available on every workstation of category **OUTPUT** or **OUTIN**. It is workstation dependent which of the interior styles **SOLID**, **PATTERN** and **HATCH** are available.

4.4.8 Fill area set attributes

Fill area set uses the same fill area attributes as fill area (see section 4.4.7).

Another set of aspects for fill area set permits independent control of the edges of fill area sets. The aspects, which can be either individual or bundled, are:

- EDGE FLAG**
- EDGE TYPE**
- EDGE WIDTH SCALE FACTOR**
- EDGE COLOUR INDEX**

The **EDGE FLAG** can have values **ON** or **OFF**. When the value is **OFF**, the edge is not drawn. When the value is **ON**, the edge is rendered according to the other **EDGE** attributes. Every fill area set primitive has an **EDGE INDEX** attribute, which selects the bundle from the workstation edge bundle table. For each edge aspect an aspect source flag exists that controls whether the individual aspect is bound to the fill area set or whether the value is taken from the bundle designated by the **EDGE INDEX**, as follows:

- EDGE FLAG ASF**
- EDGE TYPE ASF**
- EDGE WIDTH SCALE FACTOR ASF**
- EDGE COLOUR INDEX ASF**

The edges drawn according to the above aspects are drawn on top of the interior. Conceptually edges and interior are disjoint, in practice they may overlap. The boundary drawn for style **HOLLOW** is considered part of the interior. Consequently, the edges will have precedence over the interior when drawn, and will then be visible. The boundary associated with the style **HOLLOW** may be visible as well. Parts of edges that are clipped become invisible. Parts of **HOLLOW** interiors that become clipped will have a boundary drawn at the clipping boundary.

Graphical output

GKS-3D

4.4.9 Cell array attributes

The plane of the cell array primitive is defined by 3 points that are corners of a parallelogram (see figure 23). The results are implementation dependent if the points are co-linear.

Cell array has no attributes other than VIEW INDEX, HLHSR IDENTIFIER and PICK IDENTIFIER. However, an array of colour indices, which are pointers into the colour table, is part of the definition of a cell array.

4.4.10 Generalized Drawing Primitive attributes

Generalized Drawing Primitive(GDP) has no explicit geometric attributes. Such information may be specified in the GDP data record. The representation of the GDP at the workstation is controlled by zero or more of the sets of polyline, polylmarker, text, fill area and edge attributes (see 4.4.2). Whether bundle indices or associated individually specified attributes are used depends upon the values of the appropriate ASFs. The sets of attributes most appropriate for the specified GDP function are selected for the GDP as part of the definition of the GDP and are recorded in the workstation description table. Similarly, if a GDP is essentially a cell array, then an array of colour indices would be specified in the GDP data record.

4.4.11 Colour

In GKS-3D, colour is specified in a number of different situations. It may be an aspect of a primitive, in which case it is specified either in the bundle for that primitive or by the individual colour attribute for that primitive. It may be part of a pattern for FILL AREA, in which case an array of colours is specified, or it may be part of a primitive itself, namely CELL ARRAY, when an array of colours is also specified. In each case, the colour is specified as an index into a colour table on the workstation. On each workstation, there is one colour table into which all the colour indices point.

The size of the colour table is workstation independent but entries 0 and 1 always exist. Entry 0 corresponds to the background colour (the colour of the display after it has been cleared). Entry 1 is the default foreground colour, and entries higher than 1 correspond to alternative foreground colours.

Colours are described by a colour model together with a specification of colour coordinates in the colour space of that model. There is a colour model in effect at each workstation, specified by the 'Current Colour Model' entry in the workstation state list.

Colours are associated with a colour index by the function SET COLOUR REPRESENTATION. The function INQUIRE COLOUR REPRESENTATION can be used to determine the colour associated with a colour index. The parameters of these functions describe the colour as a specification of colour coordinates in the colour space of the current colour model.

The current colour model at a workstation is set by the control function SET COLOUR MODEL. This function selects the colour model used for interpretation of the parameters of SET COLOUR REPRESENTATION and INQUIRE COLOUR REPRESENTATION. It does not affect the colours associated with the indices. The current colour model at a workstation can be inquired by the function INQUIRE COLOUR MODEL.

Colour models 1 to 4 are predefined as RGB, CIE, HSV and HLS. Only RGB and CIE are required to be available. Colour models greater than 4 are required for registration and colour models less than 0 are implementation dependent. Those colour models and the appropriate conversion algorithms are described in Annex H. The colour models available on a workstation can be inquired by the function INQUIRE COLOUR MODEL FACILITIES.

GKS-3D

Graphical output

When colours are associated with a colour index, they are mapped onto the nearest colour available on the workstation. On some workstations it may not be possible to change the background colour, and in this case the mapping of specific colour to the nearest available for the background colour may be different to that of mapping the same colour for the foreground colours.

Some workstations are not capable of displaying colours (for example, workstations only capable of displaying colours with equal red, green, and blue intensities or workstations only capable of displaying colours which are different intensities of the same colour); these are referred to as monochrome workstations. Whether a workstation is capable of colour is recorded in the 'colour available' entry in the workstation description table. On monochrome workstations, the intensity is computed from the colour values in a workstation dependent way.

4.4.12 View index

An attribute of each primitive is the VIEW INDEX. The current value of the VIEW INDEX is bound to the primitive at creation. The index will select viewing parameters from a view table at each workstation. The VIEW INDEX is treated in the same fashion as the other bundle indices. The view representation is workstation dependent. The INSERT SEGMENT and INSERT SEGMENT 3 functions replace the VIEW INDEX, stored with an output primitive when the segment was created, by the current VIEW INDEX.

4.4.13 Hidden line/hidden surface removal (HLHSR) identifier

An attribute for each primitive is the HLHSR IDENTIFIER. The current value of the HLHSR IDENTIFIER is bound to the primitive at creation. The HLHSR IDENTIFIER will be interpreted at the workstation in an implementation dependent manner.

The INSERT SEGMENT and INSERT SEGMENT 3 functions replace the HLHSR IDENTIFIER, stored with an output primitive when the segment was created, by the current HLHSR IDENTIFIER.

GKS-3D

- polyline representation (see 4.4)
- polymarker representation (see 4.4)
- text representation (see 4.4)
- fill area representation (see 4.4)
- edge representation (see 4.4)
- pattern representation (see 4.4)
- colour model (see 4.4)
- colour representation (see 4.4)
- deferral state (see 4.5.3)
- HLHSR mode (see 4.5.7)
- workstation transformation (see 4.6.3)
- view representation (see 4.6.6)

Figure 2 (see 4.4) illustrates the binding of the workstation attributes.

A workstation of category INPUT has at least one logical input device, but no output capability.

A workstation of category OUTIN has the characteristics of both an OUTPUT and INPUT workstation. In addition, the existence of a workstation in this category in a GKS-3D implementation gives rise to additional requirements regarding logical input devices (see 4.8.1).

The last three categories WISS, MO and MI are special GKS-3D facilities that provide a means for temporarily or permanently storing graphical information. They are treated as workstations for the purposes of control, but otherwise have quite different characteristics (see 4.7.5 and 4.9).

Clause 5 of annex A gives a complete listing of all GKS-3D functions which apply directly or indirectly to each category of workstation.

Actual workstations may provide more capabilities than those listed in the workstation description table. These cannot be used by GKS-3D. However, if the workstation itself provides sufficient intelligence, the additional capabilities may be accessed via the GENERALIZED DRAWING PRIMITIVE or ESCAPE functions, or used locally by the workstation operator. As an example, if a workstation has two display spaces, the operator may switch locally from one to the other without notifying GKS-3D or the application program. More than one display space can be controlled by GKS-3D only by defining a separate workstation for each display space.

4.5.2 Selecting a workstation

The application program references a workstation by means of a workstation identifier. Connection to a particular workstation is established by the function OPEN WORKSTATION, which associates the workstation identifier with a workstation type and a connection identifier (for example, a unit number in FORTRAN). The current state of each open workstation is kept in a workstation state list. Segment manipulation and input can be performed on all open workstations. Output primitives are sent to, and segments are stored on, all active workstations and no others; an open workstation is made active by the function ACTIVATE WORKSTATION.

An active workstation is made inactive by the function DEACTIVATE WORKSTATION; an open workstation is closed by the function CLOSE WORKSTATION.

The following sequence of functions illustrates workstation selection:

GKS-3D

4.5 Workstations

4.5.1 Workstation characteristics

GKS-3D is based on the concept of abstract graphical workstations. These provide the logical interface through which the application program controls physical devices. Certain special workstations provide facilities for the storage and exchange of graphical information.

For every type of workstation present in a GKS-3D implementation (except for the special workstations), there exists a workstation description table which describes the capabilities and characteristics of the workstation. The application program can inquire which capabilities are available and adapt its behaviour accordingly. If a capability is requested that a particular workstation does not provide, a standard error reaction is defined. Certain minimal capabilities of a workstation are detailed in 4.10.

An abstract graphical workstation with maximum capabilities

- a) has one addressable display space of fixed resolution;
- b) allows only rectangular parallelepiped display spaces (the display space does not consist of a number of separate parts);
- c) permits the specification and use of smaller display spaces than the maximum while guaranteeing that no display image is generated outside the specified display space;
- d) supports several linetypes, text fonts, character sizes, etc., to allow output primitives to be drawn with different aspects;
- e) has one or more logical input devices for each input class;
- f) permits REQUEST, SAMPLE and EVENT type input;
- g) allows logical input devices to be set in REQUEST, SAMPLE or EVENT mode independently of each other;
- h) stores segments and provides facilities for changing and manipulating them;
- i) permits definitions of several transformations that support 3D viewing transformations.

In practice, the workstation is not necessarily equipped with all of these capabilities.

Each workstation has a type. Each workstation type falls into one of six categories:

OUTPUT	Output
INPUT	Input
OUTIN	Output and input
WISS	Workstation Independent Segment Storage(WISS)
MO	GKS-3D Metafile (GKSM) output
MI	GKSM input

A workstation of category OUTPUT has only output capabilities. It can display all output primitives, with the possible exception of the GENERALIZED DRAWING PRIMITIVE which is optional. Minimal requirements for displaying TEXT and FILL AREA primitives are listed in 4.4, and for CELL ARRAY in 5.3.

GKS-3D allows the appearance of output primitives to vary between workstations, thus allowing advantage to be taken of their differing capabilities. The facilities which allow this variation are:

OPEN WORKSTATION (N1,comid1,workstation type A);
 OPEN WORKSTATION (N2,comid2,workstation type B);
 ACTIVATE WORKSTATION (N1);

Output functions;
 Input functions; {generated only on N1}
 {possible on N1,N2}

ACTIVATE WORKSTATION (N2);

Output functions; {generated on N1,N2}

DEACTIVATE WORKSTATION (N1);

Output functions;
 Input functions; {generated only on N2}
 {possible on N1,N2}

CLOSE WORKSTATION (N1);
 DEACTIVATE WORKSTATION (N2);
 CLOSE WORKSTATION (N2);

4.5.3 Deferring picture changes

It is desirable that the display of a workstation reflects, as far as possible, the actual state of the picture as defined by the application program. However, to use the capabilities of a workstation efficiently, GKS-3D allows a workstation to delay, for a certain period of time, the actions requested by the application program. During this period, the state of the display may be undefined.

The function SET DEFERRAL STATE allows the application program to choose that deferral state which takes into account the capabilities of the workstation and the requirements of the application program. Two attributes are defined for this purpose. Deferral mode controls the time at which output functions have their visual effects. Implicit regeneration controls the time at which picture changes have their visual effects: picture changes in general imply an alteration not just an addition to the picture.

The concept of deferral refers only to visible effects of GKS-3D functions. Effects on the segment storage or on the state of the workstation are (conceptually) not deferred.

Deferral mode controls the possible delaying of output functions: for example, data sent to a device may be buffered to optimize data transfer. The values of deferral mode (in increasing order of delay) are:

a) ASAP: The visual effect of each function will be achieved on the workstation As Soon As Possible (ASAP). GKS-3D ensures that the actions necessary to achieve this visual effect are initiated before control is returned to the application program, but, owing to possible delays not under the influence of GKS-3D, the actions are not necessarily completed before control is returned.

b) BNIG: The visual effect of each function will be achieved on the workstation Before the Next Interaction Globally (BNIG), i.e. before the next interaction with a logical input device gets underway on any workstation (see 4.8.2). If an interaction on any workstation is already underway, the visual effect will be achieved as soon as possible.

GKS-3D

c) BNIL: The visual effect of each function will be achieved on the workstation Before the Next Interaction Locally (BNIL), i.e. before the next interaction with a logical input device gets underway on that workstation (see 4.8.2). If an interaction on that workstation is already underway, the visual effect will be achieved as soon as possible.

d) ASTI: The visual effect of each function will be achieved on the workstation At Some Time (ASTI).

Deferral applies to the following functions that generate output:

POLYLINE 3
 POLYLINE
 POLYMARKER 3
 POLYMARKER
 TEXT 3
 TEXT
 FILL AREA 3
 FILL AREA
 FILL AREA SET 3
 FILL AREA SET
 CELL ARRAY 3
 CELL ARRAY
 GENERALIZED DRAWING PRIMITIVE 3
 GENERALIZED DRAWING PRIMITIVE
 INSERT SEGMENT 3
 INSERT SEGMENT
 ASSOCIATE SEGMENT WITH WORKSTATION
 COPY SEGMENT TO WORKSTATION
 INTERPRET ITEM

For none of the possible values of deferral mode is it mandatory for an implementation to delay the visual effect of output functions. If the application program requires a delay, it can achieve this using the segment storage facility and the visibility attribute. This restriction means that the buffer for deferred actions can be chosen in an implementation dependent manner.

Certain functions can be performed immediately on some workstations, but on other workstations imply a regeneration of the whole picture to achieve their effect. For example, an implicit regeneration is necessary when picture changes require new paper to be put on a plotter. The entries 'dynamic modification accepted' in the workstation description table indicate which changes

e) lead to an implicit regeneration (IRG);

f) can be performed immediately (IMM).

If changes can be performed immediately, those changes may affect primitives outside segments in addition to those inside segments. If regeneration occurs, all primitives outside segments will be deleted from the display space.

An implicit regeneration is equivalent to an invocation of the function REDRAW ALL SEGMENTS. Its possible delay is controlled by the implicit regeneration mode, a single entry in the workstation state list. The values of implicit regeneration mode are

Workstations

- g) **SUPPRESSED:** Implicit regeneration of the picture is suppressed, until it is explicitly requested: the entry 'new frame necessary at update' is set to YES.
- h) **ALLOWED:** Implicit regeneration of the picture is allowed.
- An implicit regeneration is made necessary, if any of the following occur:
- i) if the functions listed below have a visible effect on the display image of the respective workstation:

- 1) if the 'dynamic modification accepted' entry in the workstation description table is IRG (implicit regeneration necessary) for the specified representation:

SET POLYLINE REPRESENTATION
 SET POLYMARKER REPRESENTATION
 SET TEXT REPRESENTATION
 SET FILL AREA REPRESENTATION
 SET EDGE REPRESENTATION
 SET PATTERN REPRESENTATION
 SET COLOUR REPRESENTATION
 SET VIEW REPRESENTATION 3

- 2) if the 'dynamic modification accepted' entry in the workstation description table is IRG for the workstation transformation:

SET WORKSTATION WINDOW 3
 SET WORKSTATION WINDOW
 SET WORKSTATION VIEWPORT 3
 SET WORKSTATION VIEWPORT

- 3) if the 'dynamic modification accepted' entry in the workstation description table is IRG for segment priority and this workstation supports segment priority:

- if primitives are added to open segment overlapping a segment of higher priority:

POLYLINE 3
 POLYLINE
 POLYMARKER 3
 POLYMARKER
 TEXT 3
 TEXT
 FILL AREA 3
 FILL AREA
 FILL AREA SET 3
 FILL AREA SET
 CELL ARRAY 3
 CELL ARRAY
 GENERALIZED DRAWING PRIMITIVE 3
 GENERALIZED DRAWING PRIMITIVE
 INSERT SEGMENT 3
 INSERT SEGMENT

(since only segments have priority, primitives outside segments do not make an implicit regeneration necessary.)
 if the complete execution of one of the following actions would be affected by segment priority:

Workstations

GKS-3D

DELETE SEGMENT
 DELETE SEGMENT FROM WORKSTATION
 ASSOCIATE SEGMENT WITH WORKSTATION
 SET SEGMENT TRANSFORMATION 3
 SET SEGMENT TRANSFORMATION
 SET VISIBILITY
 SET SEGMENT PRIORITY

- 4) if the 'dynamic modification accepted' entry in the workstation description table is IRG for segment transformation:

SET SEGMENT TRANSFORMATION 3
 SET SEGMENT TRANSFORMATION

- 5) if the 'dynamic modification accepted' entry in the workstation description table is IRG for 'visibility (visible → invisible)':

SET VISIBILITY (INVISIBLE)

- 6) if the 'dynamic modification accepted' entry in the workstation description table is IRG for 'visibility (invisible → visible)':

SET VISIBILITY (VISIBLE)

- 7) if the 'dynamic modification accepted' entry in the workstation description table is IRG for highlighting:

SET HIGHLIGHTING

- 8) if the 'dynamic modification accepted' entry in the workstation description table is IRG for delete segment:

DELETE SEGMENT
 DELETE SEGMENT FROM WORKSTATION

- 9) if the 'dynamic modification accepted' entry in the workstation description table is IRG for HLHSR mode:

SET HLHSR MODE

- 10) if the HLHSR MODE is such that the workstation will perform HLHSR and the operating state is SGOP:

POLYLINE 3
 POLYLINE
 POLYMARKER 3
 POLYMARKER
 TEXT 3
 TEXT
 FILL AREA 3
 FILL AREA
 FILL AREA SET 3
 FILL AREA SET
 CELL ARRAY 3
 CELL ARRAY
 GENERALIZED DRAWING PRIMITIVE 3
 GENERALIZED DRAWING PRIMITIVE
 INSERT SEGMENT 3
 INSERT SEGMENT

j) if any of the above situations occurs as a result of INTERPRET ITEM.

An implicit regeneration has to be done (including deletion of primitives outside segments) only if one of the functions listed causes a visible effect on the display; for example, if an invisible segment is deleted, a regeneration need not be done. However, an implementation is allowed to perform an implicit regeneration in any of the cases listed above.

Deferred actions can be made visible at any time by the use of the UPDATE WORKSTATION function or by an appropriate change of the deferral state.

4.5.4 Clearing the display space

Two capabilities for clearing the display space are recognized, namely:

- clear the display space even if it is empty;
- ensure that the display space is clear without clearing the display space needlessly.

The second capability means that the display space is only cleared when needed: this would normally be when the display space is not clear, (i.e. when the 'display space empty' entry in the workstation state list is NOTEMPTY). The entry 'display space empty' in the workstation state list is set to NOTEMPTY if output is sent to the device. It may be set to NOTEMPTY even if output does not appear on the display space (for example, a GDP primitive which has been clipped at the device to non-existence).

Both capabilities for clearing the display space are available to the user through the function CLEAR WORKSTATION. The second capability is also used in UPDATE WORKSTATION and REDRAW ALL SEGMENTS ON WORKSTATION.

4.5.5 Elimination of primitives outside segments

Elimination of primitives outside segments occurs in the following situations:

- if the following GKS-3D functions are invoked:

CLEAR WORKSTATION
 REDRAW ALL SEGMENTS ON WORKSTATION
 UPDATE WORKSTATION

if the parameter update regeneration flag is PERFORM and if 'new frame action necessary at update' in the workstation state list is YES;

SET DEFERRAL STATE

if 'implicit regeneration mode' is ALLOWED and 'new frame action necessary at update' is YES;

- if an implicit regeneration is made necessary (see 4.5.3) and 'implicit regeneration mode' is ALLOWED;
- if any of the above situations occurs as a result of INTERPRET ITEM.

4.5.6 Sending messages to a workstation

The MESSAGE function allows a character string to be sent to a workstation. The application program has no control over the position and appearance of the character string and an implementation is allowed to place the string on a device distinct from, but associated with, the workstation. The rules to be followed by an implementation are stated in 5.2.

4.5.7 Hidden line/hidden surface removal

Hidden line/hidden surface removal (HLHSR) is the process of determining which parts of primitives are invisible as a result of being obscured by other primitives when observed from the PRP or along the projection direction (see 4.6.6). In GKS-3D, the method by which these parts are calculated is implementation dependent. Moreover, implementation of any HLHSR capabilities is not mandatory.

HLHSR is controlled on a workstation by workstation basis, by the workstation attribute HLHSR MODE and, on a primitive by primitive basis, by a global attribute HLHSR IDENTIFIER. Both attributes are specified as integers the meaning of which are determined in a workstation and implementation dependent manner. The attributes may not be used for other purposes than HLHSR. The default value for both attributes is 0. Where appropriate, 0 should be the most convenient action for that workstation.

Adding a primitive may cause an implicit regeneration if HLHSR is being performed. Primitives outside segments do not take part in HLHSR calculations. Primitives in segments that have the segment attribute visibility set to INVISIBLE cannot obscure other primitives. Primitives that are not visible as a result of HLHSR are not detectable.

The INSERT SEGMENT and INSERT SEGMENT 3 functions replace the HLHSR IDENTIFIER bound to a primitive by the current HLHSR IDENTIFIER.

HLHSR removal takes precedence over segment priority.

Coordinate systems and transformations

GKS-3D

GKS-3D

Coordinate systems and transformations

4.6 Coordinate systems and transformations

4.6.1 Normalization transformations

In GKS-3D, the application programmer can compose his graphical picture from separate parts each of which, conceptually, is defined with its own world coordinate system (WC). The relative positioning of the separate parts is defined by having a single normalized device coordinate space (NDC) onto which all the defined world coordinate systems are mapped. A set of normalization transformations define the mappings from the world coordinate systems onto the single normalized device coordinate space, which can be regarded as a workstation independent abstract viewing space. This normalized picture can be stored and manipulated via the segment mechanism; it can also be stored on a metafile.

For output, a single normalization transformation is current at any one time and this is used to transform world coordinates specified, for example in output primitives and geometric attributes, into normalized device coordinates.

A normalization transformation is specified by defining the limits of a volume in the world coordinate system (window) which is to be mapped onto a specified volume of the normalized device coordinate space (viewport). Window and viewport limits specify rectangular parallelepipeds with edges parallel to the coordinate axes in WC and NDC. The volumes include their boundaries. The normalization transformation performs a mapping from WC onto NDC that includes translation and differential scaling with positive scale factors for the three axes. The scale factor may be zero for the z-axis.

Although NDC space conceptually extends to infinity, the part of NDC space in which the viewport needs to be located and that can be viewed at a workstation is the closed range $[0,1] \times [0,1] \times [0,1]$. In addition, an implementation may support only a restricted range of NDC space. However, this range is always sufficiently greater than the $[0,1] \times [0,1] \times [0,1]$ cube that useful effects of INSERT SEGMENT and INSERT SEGMENT 3 can be achieved. In particular, NDC coordinates in the range $[-7,7] \times [-7,7] \times [-7,7]$ are always handled.

Each normalization transformation is identified by a transformation number which is an integer between 0 and an implementation dependent value n which can be inquired from the GKS-3D state list. The normalization transformation with transformation number 0 is the unity transformation which maps $[0,1] \times [0,1] \times [0,1]$ in world coordinates to $[0,1] \times [0,1] \times [0,1]$ in normalized device coordinates. It cannot be changed.

Initially, all other normalization transformations are set to a default transformation which is the same as transformation number 0. Different transformations can be specified at any time when GKS-3D is open. Since GKS-3D provides a number of different normalization transformations, it is possible for the application program to specify them prior to outputting the graphical picture. The separate parts of the picture are output by selecting a particular normalization transformation before outputting the associated graphical primitives. However, specifying a normalization transformation, while the graphical output is taking place, is allowed.

A normalization transformation may be selected by SELECT NORMALIZATION TRANSFORMATION, and it will be used for all output until another is selected. By default, normalization transformation 0 is selected.

4.6.2 Clipping

The viewport of a particular normalization transformation is used to define a clipping volume, as well as, with the window, specifying the normalization transformation. When the viewport of the current normalization transformation is set or when a normalization transformation is selected, the 'clipping volume' entry in the GKS-3D state list is set to the resulting viewport of the current normalization transformation. Clipping to this clipping volume can either be enabled or disabled. There is a single global switch (the clipping indicator) which defines whether or not the clipping volume is to be used for clipping.

Clipping does not take place when the normalization transformation is performed but is delayed until the output primitives are to be displayed in the display space of a workstation. Output primitives stored in segments have their coordinates transformed to NDC and the associated clipping volume and clipping indicator stored with the primitives. The INSERT SEGMENT and INSERT SEGMENT 3 functions allow the clipping volume and clipping indicator in the GKS-3D state list to replace the clipping volume and clipping indicator stored with an output primitive when the segment was defined.

Primitives sent to a workstation of category MO are not clipped.

4.6.3 Workstation transformations

Points in NDC space are transformed into Normalized Projection Coordinates (NPC) space by the viewing operations described in 4.6.6. NPC space can be regarded as a workstation independent abstract viewing space. Each open workstation can select independently some part of NPC space in the range $[0,1] \times [0,1] \times [0,1]$ to be displayed somewhere in the workstation display space. A particular workstation transformation is a mapping from NPC space onto the device coordinates (DC) space for that workstation.

The units of device coordinates are metres on a device capable of producing a precisely scaled image (for example, on most plotters) and appropriate workstation dependent units otherwise (for example, on a display unit with unknown monitor size). In either case the device coordinate system maps onto the display space in the following way:

- a) the DC origin is at the bottom left corner furthest from the observer of the display space;
- b) the device coordinate units are related to the display space in such a way that a cube in device coordinates appears as a cube on the display space (this is trivially true if device coordinate units are metres);
- c) x, y and z increase to the right, upwards and towards the observer respectively.

On some devices, device coordinate units do not coincide with addressable units, for example if the addressable units do not satisfy the above conditions.

The size of the display space in device coordinate units is recorded in the workstation description table.

The workstation transformation is a uniform mapping from NPC onto DC for x and y and thus performs translation and *equal* scaling with a positive scale factor for these two axes. Thus picture composition can be achieved using the normalization transformations whereas the workstation transformation allows different aspects of the composed picture to be viewed on different workstations. For example, a drawing could be output to a plotter at the correct scale and simultaneously some part of the drawing could be displayed on the full display space of an interactive terminal.

The workstation transformation can be specified at any time after the workstation has been opened.

However, actually changing the workstation transformation may cause an implicit regeneration of the picture.

A workstation transformation is specified by defining the limits of a volume in the normalized projection coordinate system within the range $[0,1] \times [0,1] \times [0,1]$ (workstation viewport) which is to be mapped onto a specified volume of the device coordinate space (workstation viewport). Workstation window and workstation viewport limits specify rectangular parallelepipeds with edges parallel to the coordinate axes in NPC and DC. The volumes include their boundaries. To ensure that no output outside the workstation window is displayed, GKS-3D clips at the workstation window boundaries, and this clipping cannot be disabled.

As the workstation window is defined within the NPC range $[0,1] \times [0,1] \times [0,1]$, this ensures that the only part of NPC space that can be viewed on any workstation lies in the range $[0,1] \times [0,1] \times [0,1]$. If the workstation window and workstation viewport have different aspect ratios for x, y, and z, the scaling specified would be different on each axis if the workstation window was mapped onto the workstation viewport in its entirety. To ensure optimum results on the display space, the workstation transformation maps the workstation window onto the largest rectangular parallelepiped that can fit within the workstation viewport such that:

- d) aspect ratio in x and y is preserved;
- e) the lower left hand corner of the workstation window closest to 0 in z is mapped to the lower left hand corner of the workstation viewport furthest from the observer;
- f) the z extent of the workstation window is mapped to the entire z extent of the workstation viewport.

Thus, space is left unused at the top or right side of the workstation viewport if the aspect ratios of the workstation window and workstation viewport are different.

All workstation transformations are set by default to map NPC space $[0,1] \times [0,1] \times [0,1]$ onto the whole of the workstation display space. If the display space is not a cube, the same rules as above apply to determine the scaling on each axis.

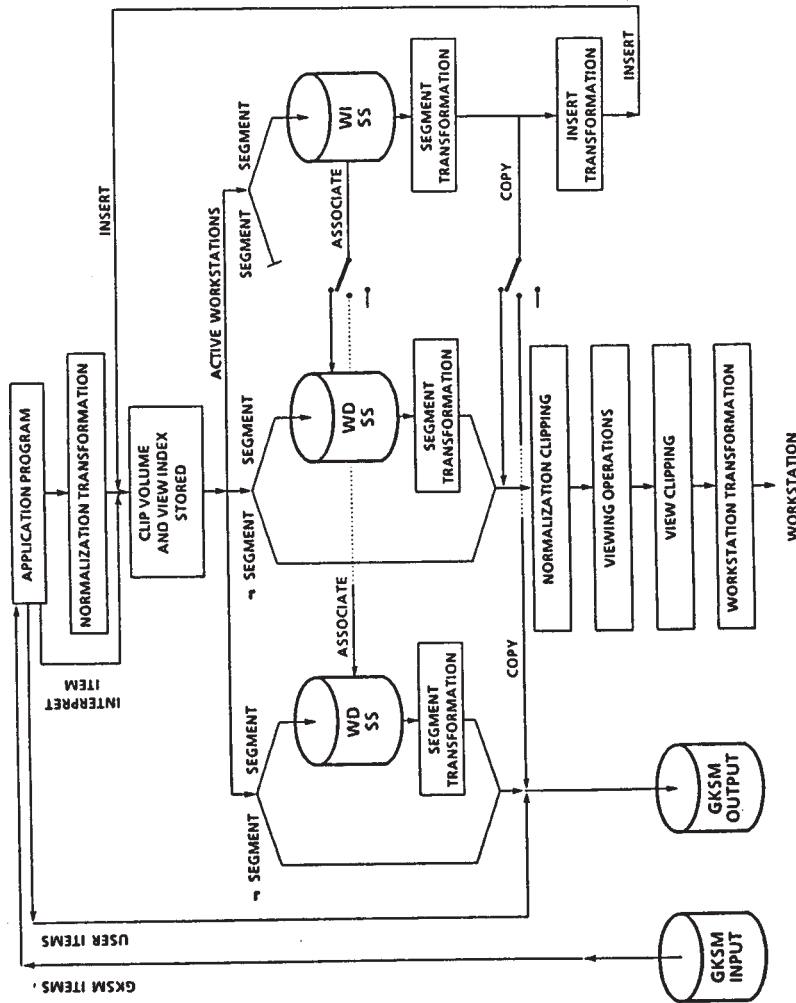


Figure 10 - Data flow chart for graphical output in GKS-3D

Coordinate systems and transformations

GKS-3D

Workstation transformations can be changed by SET WORKSTATION WINDOW, SET WORKSTATION WINDOW 3, SET WORKSTATION VIEWPORT, or SET WORKSTATION VIEWPORT 3. As a specification of the workstation transformation may be deferred (see 4.5.3), these two functions only set the 'requested workstation window' and 'requested workstation viewport' entries in the workstation state list. The 'current workstation window' and 'current workstation viewport' entries continue to hold the previously set transformation parameters. When the display is updated, the current values are set to the requested values.

A complete data flow chart for graphical output is given in figure 10. It should be noted that all three coordinate systems (WC, NPC and DC) are three dimensional Cartesian coordinate systems.

4.6.4 Transformation of locator input

The application programmer requires LOCATOR input to define a position in the most appropriate world coordinate system currently defined by the set of normalization transformations.

This is achieved by first transforming the input data from DC to NPC by the inverse workstation transformation which is in effect when LOCATOR input is generated. LOCATOR input can only be obtained from positions within the part of the current workstation viewport into which the current workstation window is mapped (note that this is a subset of the workstation viewport whenever the aspect ratio of the workstation viewport and workstation window differ). Thus, LOCATOR input always defines a position in the NPC range $[0,1] \times [0,1] \times [0,1]$.

The position in NPC space needs to be transformed from NPC to NDC by the inverse of the viewing transformation in one of the view representations. Each view representation has associated with it a view transformation input priority which is only relevant to LOCATOR and STROKE input. View representations are ordered in a list defined by the view transformation input priority. When a GKS-3D workstation is opened, a workstation-dependent number of view representations are initialized, and their view transformation input priorities are set with view representation 0 given the highest priority, view representation 1 the next highest and so on. Changing the view transformation input priority of any view transformation is allowed at any time. The LOCATOR input position in NPC space is compared with the view clipping limits of the view representations to find the view representation with the highest view transformation input priority whose view clipping limits contain the LOCATOR position.

The LOCATOR position is transformed by the inverse of this view transformation to the associated NDC.

To return to the application program a position in world coordinates, the position in NDC space needs to be transformed from NDC to WC by the inverse of one of the normalization transformations. Each normalization transformation has associated with it a viewport input priority which is only relevant to LOCATOR and STROKE input. Normalization transformations are ordered in a list defined by the viewport input priority. At GKS-3D initialization, an implementation defined number of normalization transformations are initialized to have window and viewport set to the cube $[0,1] \times [0,1] \times [0,1]$ and their viewport input priorities are set relative to the transformation number with transformation number 0 given the highest priority, transformation number 1 the next highest and so on. Changing the viewport input priority of any normalization transformation is allowed at any time.

The LOCATOR input position in NDC space is compared with the viewports of the normalization transformations, to find the normalization transformation with the viewport which has the highest viewport input priority and contains the LOCATOR position. The LOCATOR position is transformed by the inverse of this normalization transformation to the associated WC. This

GKS-3D

Coordinate systems and transformations

LOCATOR position is returned to the application program in WC together with the number of the normalization transformation and view index used.

If the point, after being transformed by the inverse viewing transformation, lies outside all normalization transformation viewports, normalization transformation 0 is used.

A data flow chart for LOCATOR input is given in figure 11.

As normalization transformation and view representation number 0 are given the highest input priority initially, LOCATOR input is effectively returned in WC equivalent to NPC until either a normalization transformation is defined with an input priority greater than that of normalization transformation number 0 or a view representation is defined with an input priority greater than that of view representation number 0. If a normalization transformation or view representation is no longer required for mapping LOCATOR input back to WC, it can effectively be hidden by reassigning it an input priority lower than that of normalization transformation or view representation number 0.

Changing the priority of view transformation and normalization transformation number 0 is allowed.

In an event report, generated by a LOCATOR device in EVENT mode, the DC position is transformed to the appropriate WC position before the event report is placed on the input queue. These transformations may be performed while the normalization and workstation transformations are being changed; thus, there is a race condition. The implementation has therefore to treat the transformations as resources to be allocated and deallocated between the competing processes.

Between placing on the input queue and the execution of AWAIT EVENT which removes the LOCATOR position from the queue, it is possible for the normalization transformation and the workstation transformation to be changed by the application program. To ensure that the DC position located is equivalent to the WC position retrieved from the input queue, it is advisable for the application program not to change a transformation while a LOCATOR is in EVENT mode.

4.6.5 Transformation of stroke input

Similar considerations apply to transformation of STROKE input as apply to LOCATOR input, with the complication that more than one point is involved.

When each point of a stroke is generated, the coordinates of the point are transformed from DC to NPC by the inverse workstation transformation then in effect. STROKE input can only be obtained from positions within the part of the current workstation viewport into which the current workstation window is mapped (analogous to LOCATOR input). Thus STROKE input always consists of points in the NPC range $[0,1] \times [0,1] \times [0,1]$.

The position in NPC space needs to be transformed from NPC to NDC by the inverse of one of the viewing transformations.

The STROKE points in NPC space are compared with the projection viewports of the view transformations to find the view with the projection viewport that has the highest view transformation input priority and contains all the STROKE points. The STROKE points are transformed by the inverse of this view transformation to the associated NDC.

To return to the application program points in world coordinates, the points in NDC space need to be transformed from NDC to WC by the inverse of one of the normalization transformations. The STROKE points in NDC space are compared with the viewports of the normalization transformations, to find the normalization transformation with the viewport which has the highest viewport input priority and contains all of the points. The STROKE points are then transformed by the inverse of this normalization transformation and returned to the application program in WC together with the number of the normalization transformation and view index used.

If the STROKE device is in SAMPLE mode, the normalization and view transformation used may vary between successive samples.

In EVENT mode, there is a similar race condition to that applying to LOCATOR input. Between placing an event report on the input queue and the execution of AWAIT EVENT which removes the STROKE event from the queue, it is possible for the normalization transformation and the workstation transformation to be changed by the application program. To ensure that the DC points input by the operator are equivalent to the WC points retrieved from the input queue, it is advisable for the application program not to change transformations while a STROKE device is in EVENT mode.

4.6.6 Viewing

Viewing is the mechanism whereby coordinates in the Normalized Device Coordinate system are transformed to Normalized Projection Coordinates. Figure 12 shows the position of viewing in the graphical output pipeline. The view index attribute of a primitive is used to select an entry in a view table at each workstation. Figure 13 shows the effect on primitives of having a view index attribute of 2 displayed on multiple workstations. The view table entry specifies a view orientation matrix, a view mapping matrix, view clipping limits in NPC, and three clipping indicators. Figure 14 shows the three stages of viewing controlled by a view table entry. The two matrices are 4×4 homogeneous coordinate transformation matrices. The view clipping limits are 6 values: XMIN,

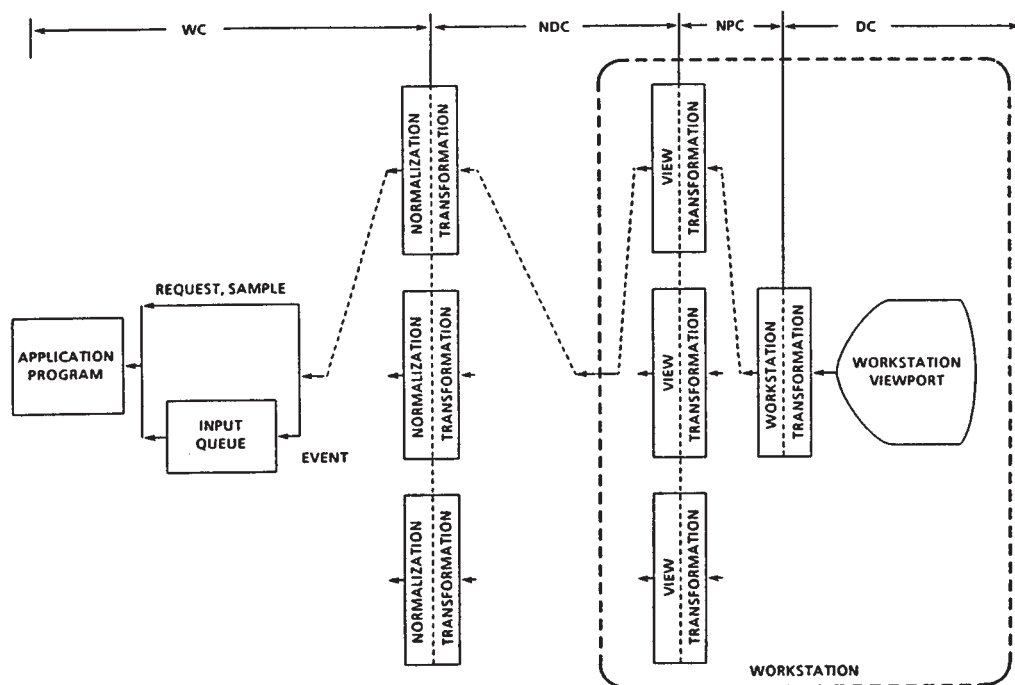


Figure 11 - Data flow chart for locator input

Coordinate systems and transformations

GKS-3D

XMAX, YMIN, YMAX, ZMIN, ZMAX defining a rectangular parallelepiped oriented with edges parallel to the NPC axes. The three clipping indicators are of a two-valued enumerated data type and comprise the front clipping indicator which controls clipping against the $z = ZMAX$ plane, the back clipping indicator which controls clipping against the $z = ZMIN$ plane, and the $x-y$ indicator which controls clipping against $x = XMIN$, $x = XMAX$, $y = YMIN$ and $y = YMAX$ planes.

The purpose of the view orientation matrix is to transform homogeneous coordinate positions in NDC into coordinates relative to the View Reference Coordinate system (VRC). The axes of the VRC system are called U, V, and N.

The purpose of the view mapping matrix is to transform points in VRC to points in the Normalized Projection Coordinate system (NPC). This transformation is typically a parallel or perspective transformation.

The purpose of the view clipping limits is to specify the region of NPC space in which visible data may appear, provided clipping is turned on and ignoring clipping defined by the workstation window. The purpose of the clipping indicators is to control whether the planes of the clipping limits are active or inactive, so that the potentially viewable portion of NPC may extend beyond the view clipping limits.

If the $x-y$ clipping indicator is NOCLIP, the region of NPC space in which visible data may appear is unlimited in x and y . If the front clipping indicator is NOCLIP, the region of NPC space in which visible data may appear is unlimited in the positive Z direction. If the back clipping indicator is NOCLIP, the region of NPC space in which visible data may appear is unlimited in the negative Z direction.

All or part of the viewing operations may be accomplished by other technological means at the workstation, in a manner not under application control. For example, workstations with holographic capabilities may use human perception to accomplish perspective. In this case it is workstation dependent whether all, part, or none of the viewing mechanism is used in a workstation.

A workstation defined number of views can be stored in each workstation. They are numbered consecutively starting with 0. In each view table there is a special entry, having index 0, that cannot be modified. It is predefined and contains the following information:

view orientation matrix:	identity, i.e. 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1
view mapping matrix:	identity, i.e., 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1
view clipping limits:	0, 1, 0, 1, 0, 1
$x-y$ clipping indicator:	CLIP
front plane clipping indicator:	CLIP
back plane clipping indicator:	CLIP

GKS-3D

Coordinate systems and transformations

The effect of view table entry 0 is to map the cube $[0,1] \times [0,1] \times [0,1]$ in VRC to the cube in NPC: $[0,1] \times [0,1] \times [0,1]$.

In GKS-3D, the predefined view table entries are initialized to values in the workstation description table, the others are initialized to the same values as view table entry 0. View table entries with index higher than 0 can be modified by the function SET VIEW REPRESENTATION 3. This function takes a workstation identifier and a view index in addition to the values to be stored in the view table entry.

Since the visual effect of setting a view table entry can be deferred, the function SET VIEW REPRESENTATION 3 only sets the 'requested' entries in the view table in the workstation state list. The 'current' entries in the view table in the workstation state list continue to hold their previous values. When the display is next updated the current values are set to the requested values.

View table entries are ordered in a list defined by the view transformation input priority. The view table entry with view index zero has the highest view transformation input priority initially.

In GKS-3D, the function SET VIEW INDEX sets the value of the 'current view index' in the GKS-3D state list. This index is bound to all output primitives.

4.6.7 Viewing utility functions

Utility functions are provided that compute 4×4 matrices suitable for use as view orientation matrices and view mapping matrices. The parameters for these utility functions are based on 3D models of orientation and projection. These models explain how certain parameters are used to determine 4×4 matrices. While GKS-3D supports certain models, other models are possible and can be utilized by an application by creating its own matrices in accordance with its desired model.

Both of the utility functions described below have two output parameters, a matrix and an error indicator. If the input parameters are incorrectly or inconsistently specified, the error indicator is set to indicate what the problem is and the error handling procedure is not invoked.

In GKS-3D, the input parameters are either in WC or in NDC, depending on a 'coordinate switch'. If the 'coordinate switch' has the value WC, the parameters are first transformed by the current normalization transformation.

The 3D model for orientation provides for translation and rotation of axes. This model is supported by the function EVALUATE VIEW ORIENTATION MATRIX 3. The input parameters of this function are as follows.

The 'view reference point' in WC or NDC defines the origin of the VRC. The view reference point is typically a point on or near the object to be viewed.

The 'view plane normal' is a vector relative to the view reference point and defines the N axis of the VRC system. The N axis is the third axis of the VRC system. The plane in WC or NDC that contains the view reference point and that is perpendicular to the view plane normal is called the 'view reference plane'.

The 'view up vector' is a vector relative to the view reference point. The vector is projected onto the view reference plane via a projection parallel to the view plane normal. The projection of the view up vector onto the view reference plane determines the V axis of the VRC system. The U axis is determined such that the UVN axes form a right-handed coordinate system. Figure 17 shows the

Coordinate systems and transformations

GKS-3D

geometry of the up vector and the UVN axes.

The 3D model for view mapping provides for parallel and perspective transformation of the VRC system to the NPC system. This model is supported by the function EVALUATE VIEW MAPPING MATRIX 3. In this model a volume in VRC called the 'view volume', is mapped to a volume in NPC bounded by 'projection viewport limits'. The 4×4 matrix returned by EVALUATE VIEW MAPPING MATRIX 3 performs this mapping. Figures 15 and 16 show typical view volumes associated with parallel and perspective views, respectively. The input parameters of this function are as follows.

The 'projection viewport limits' is a rectangular parallelepiped in NPC space with edges parallel to the NPC axes. Although NPC conceptually extends beyond $[0,1] \times [0,1] \times [0,1]$, the part of NPC in which the view clipping limits must be located is the closed unit cube $[0,1] \times [0,1] \times [0,1]$. The six NPC values defining the projection viewport are referred to as XMIN, XMAX, YMIN, YMAX, ZMIN, and ZMAX. Typically the projection viewport limits and the view clipping limits will be set to the same values, but this is not mandatory.

The front plane, back plane, and view plane are planes parallel to the UV plane of the VRC system. They are specified as N coordinate values by the parameters 'front plane distance', 'back plane distance', and 'view plane distance', respectively. The front plane and the back plane contain the front and back of the view volume. Conceptually, the View Reference Coordinate system is "oriented" (since VRC coordinates result from the 'view orientation transformation'). Since the front plane should not be behind the back plane, the front plane distance shall not be less than the back plane distance.

The view window is a rectangular region on the view plane with sides parallel to the U and V axes. The view window is specified as U and V coordinate values by the 'window limits' parameter. The window limits parameter consists of four values usually referred to as UMIN, UMAX, VMIN, and VMAX.

By definition the UMIN edge is the left edge of the view window, the UMAX edge is its right edge, the VMIN edge is its bottom edge, and the VMAX edge is its top edge.

The left, right, bottom and top edges of the view window, together with the projectors through those edges, define the left, right, bottom, and top surfaces of the view volume (respectively). In this model for view mapping, the term 'projector' refers to an infinite line in VRC that is mapped (via the view mapping) to an infinite line in NPC of the form ($X = \text{constant}$, $Y = \text{constant}$, Z arbitrary). This meaning is slightly different from the meaning of 'projector' common in mathematics.

The 'projection type' parameter is an enumerated type with the values PARALLEL and PERSPECTIVE.

The 'projection reference point' (PRP) is a position in VRC space and serves to orient the projectors defining the surfaces of the view volume. If projection type is PARALLEL, the projectors are all parallel to the vector joining the PRP and the centre of the view window. The view volume is thus a parallelepiped. If projection type is PERSPECTIVE the projectors all pass through the PRP. In this case the projectors passing through the view window lie on the surface of a pair of infinite rectangular cones having their common vertex at the PRP. The view volume is thus a portion of this double cone.

There are the following restrictions on the parameters:

$$XMIN < XMAX, YMIN < YMAX, ZMIN \leq ZMAX;$$

GKS-3D

Coordinate systems and transformations

XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX must be in the range $[0,1]$, inclusive;
 $UMIN < UMAX, VMIN < VMAX$.

In addition, back plane distance shall be less than front plane distance unless ZMIN equals ZMAX in which case they may be equal. The N coordinate of the PRP shall not equal the view plane distance. For a projection type of PERSPECTIVE, the N coordinate of the PRP may not lie between the front plane distance and the back plane distance, inclusive.

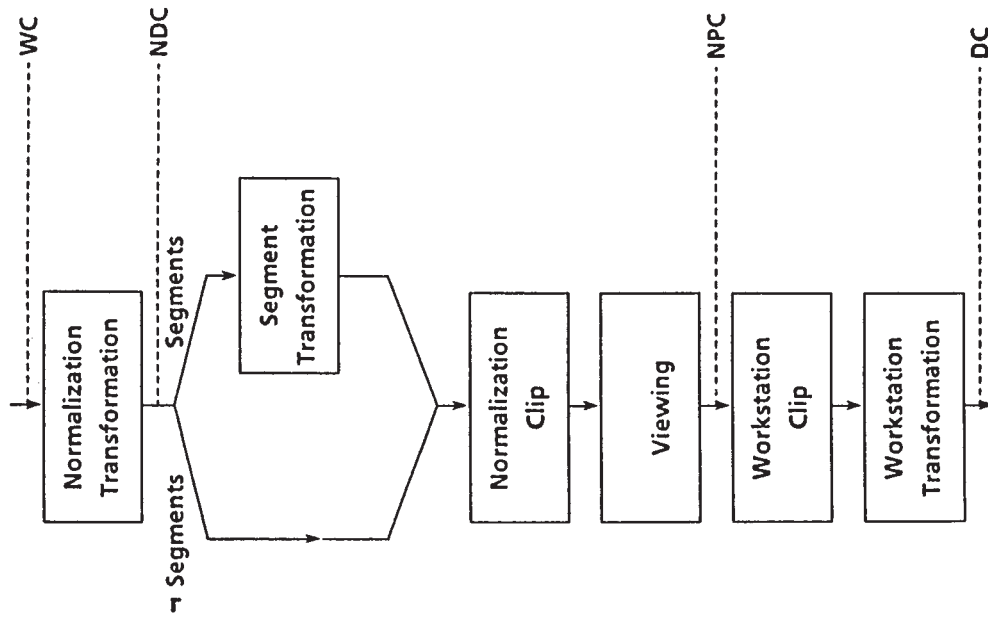


Figure 12 - The GKS-3D viewing pipeline

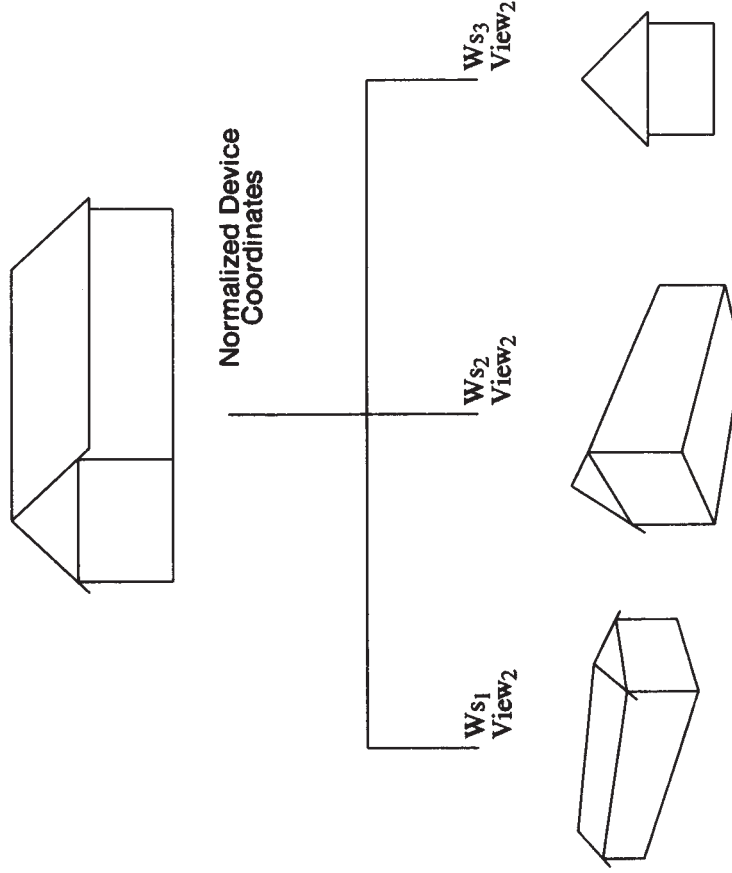


Figure 13 - Interpretation of the view index on active workstations



Figure 14 - Viewing model for view representation



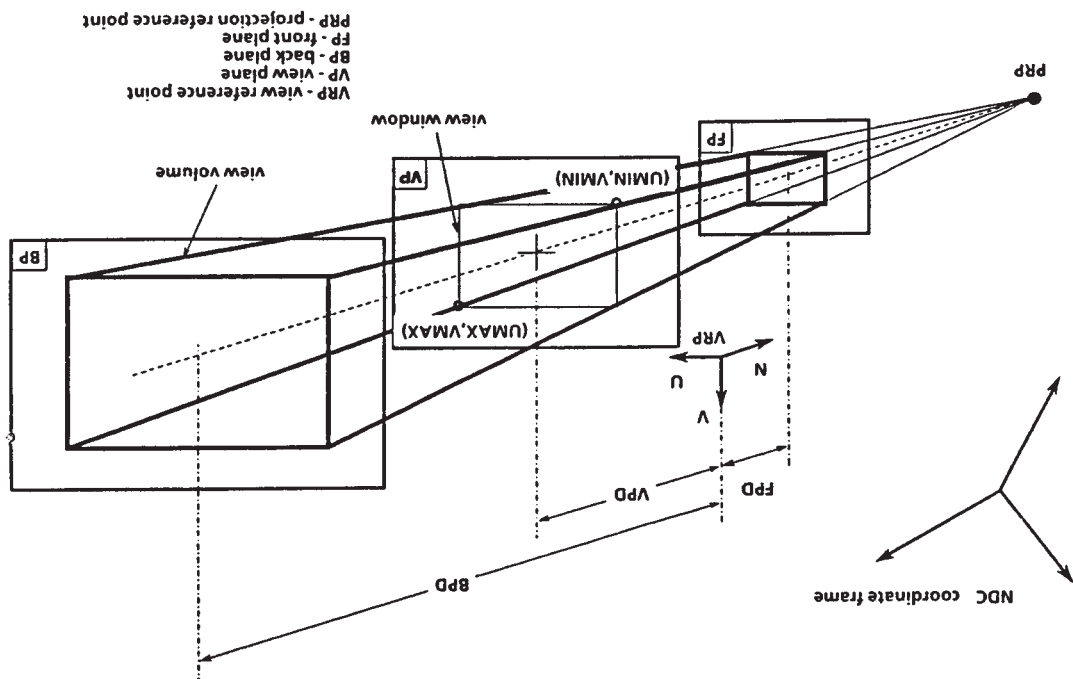
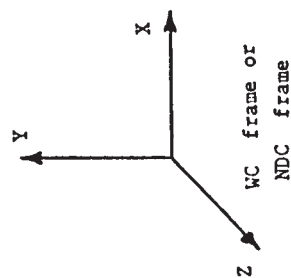


Figure 16 - Perspective projection view volume



Vector N points into the paper

Figure 17 - Formation of V axis in VRC system

Segments

GKS-3D

4.7 Segments

4.7.1 Introduction to segments

In GKS-3D, graphical output primitives may be grouped in segments as well as being created outside segments. Each segment is identified by a unique, application specified segment name. Segments may be

- a) transformed;
 - b) made visible or invisible;
 - c) highlighted;
 - d) ordered front to back, which impacts overlapping primitives;
 - e) made detectable or undetectable;
 - f) deleted;
 - g) renamed;
 - h) inserted into the open segment or into the stream of primitives outside segments (see 4.7.6).
- Only primitives contained inside segments are affected by these operations. The application program has no access to primitives outside segments once they have been created.

Every primitive within a segment has an attribute **PICK IDENTIFIER** which establishes a second level of naming. The sole function of the **PICK IDENTIFIER** is the identification of primitives; it cannot be used for manipulations. This level of naming is provided in GKS-3D to reduce the segment overhead for applications where a great number of picture parts need to be distinguished for input but the need for manipulation is less important.

Whereas segment names are unique, the same value for **PICK IDENTIFIER** can be assigned arbitrarily to single output primitives or groups of output primitives within segments, as illustrated in the following sequence of functions:

```
SET PICK IDENTIFIER(4);
CREATE SEGMENT (1);
  Output functions;
SET PICK IDENTIFIER (2);
  Output functions;
CLOSE SEGMENT;
  Output functions;
{
  (segment = 1, PICK IDENTIFIER = 4)
  (segment = 1, PICK IDENTIFIER = 2)
  (primitives not pickable)
  {PICK IDENTIFIER = 2}
  (primitives not pickable)
  {PICK IDENTIFIER = 5}
  (segment = 2, PICK IDENTIFIER = 5)
  (segment = 2, PICK IDENTIFIER = 3)
}
```

After a segment is closed, primitives in it cannot be modified nor can primitives be added to or deleted from the segment. No function is provided to extend a segment after it has been closed. Clipping indicators, clipping volumes and primitive attributes (geometric attributes, attributes controlling non-geometric aspects, and **PICK IDENTIFIER**) are stored along with primitives in segment storage. Geometrical transformations, changes of the segment attributes and changes of the

GKS-3D

Segments

workstation specific bundle and colour tables referenced from within a segment are possible. All values describing the state of a segment (i.e. name, segment attributes, and workstations active at creation time) are stored in a segment state list that GKS-3D keeps during a segment's lifetime.

Each segment is stored on all workstations active at the time the segment is created (**CREATE SEGMENT**). It can be deleted on all workstations by the **DELETE SEGMENT** function. It can be deleted on a specific workstation by the **DELETE SEGMENT FROM WORKSTATION** function. All segments stored on a specific workstation can be deleted from it by the **CLEAR WORKSTATION** function.

Segment storage on an **OUTPUT** or **OUTIN** workstation is referred to as **Workstation Dependent Segment Storage (WDSS)**. GKS-3D provides a second segment storage system for workstation independent storage of pictures at run time, **Workstation Independent Segment Storage (WISS)**; see 4.7.5 and 4.7.6). Segments cannot be moved from **WDSS** to another workstation, but from **WISS** they can.

Segments have a unique name across all segment storage. A GKS-3D implementation provides a large number of available segment names (for example 32 000).

4.7.2 Segment attributes

Segment attributes affect all the primitives in a segment. The segment attributes are:

- a) **SEGMENT**
TRANSFORMATION: (see 4.7.3);
a segment is either displayed or not;
- b) **VISIBILITY**:
a visible segment is either highlighted or not;
- c) **HIGHLIGHTING**:
if parts of segments (for example, **FILL AREA**, **CELL ARRAY**) overlap, the segment with the higher priority will be preferred,
- d) **SEGMENT PRIORITY**:
both when the segments are displayed and when they are picked;
- e) **DETECTABILITY**:
a segment can either be selected by a pick input device or it cannot.

The segment attributes are unique for each segment and do not vary on different workstations. The default segment attributes (identity transformation, visible, not highlighted, priority 0, undetectable) are assigned to a segment when it is opened. The segment attributes of any segment in existence, including the open segment, may be changed. The binding of segment attributes is shown in figure 2 (see 4.4).

Segment priority affects segments being displayed (i.e. performing segment and workstation transformations, including clipping, for each primitive of the segment). If parts of primitives overlap with others of a visible segment with higher priority, these parts may be invisible. Whether a workstation supports this feature is indicated in the workstation description table. This feature is intended to address appropriate hardware capabilities only. It is not intended to mandate shielding on non-raster displays. When primitives within a segment overlap, the implementation determines the appearance of the overlapped parts. The actual effect is listed in the documentation accompanying an implementation.

When primitives of segments overlapping each other are picked, the segment with higher priority is selected. When primitives of the same segment or of segments with equal priority overlap, the results are implementation dependent.