

3次元グラフィックスとマン・マシン・インタラクション

— GKS-3D と PHIGS —

木村文彦 (東京大学工学部)

[1] はじめに

グラフィックス基本ソフトウェア標準GKSは、1982年6月に開催されたISO/TC97/SC5/WG2の会議においてDIS¹⁾として承認され、全ての技術的検討を一応終了した。同時に、3次元機能を考えるサブグループが結成され、作業が開始された。高度なインタラクションの機能の標準化や3次元ハードウェアの利用などが目的とされていたが、「3次元」の範囲が明確でなく、作業は進展しなかった。

1983年9月のWG2の会議において、ノルウェーよりIDIGS²⁾、西独よりGKSモジュラー拡張案³⁾、オランダよりGKS-3D拡張案⁴⁾が提案された。西独案は、従来のGKSと並列に、3次元のグラフィック・データを扱う全く同様な構成の3次元GKSを考える、相互にデータ交換ができるようにしたものであった。オランダ案は、従来のGKSのポリ・プロセサとして3次元機能も位置付け、3次元データも2次元へ投影したものが従来のGKSへ供給される、とした。これらがGKSとの適合性を重視したのに対し、ノルウェー案は3次元viewing pipelineを基本としており、GKSとの不適合な箇所が散見されるものであった。

一方、アメリカはANSIで検討中のPHIGS⁵⁾を参考資料として提出し、

	ISO WG2	GKS-3D	PHIGS
1982	June Approve GKS as DIS. 3D Subgroup.		July Project Initiation.
1983	Sept GKS-3D started.	Dec First Draft.	Nov Base-line documents. (Rev. 03)
1984	June GKS-3D fin. (PHIGS ??)	Mar First Comments. May Second Draft. June Issue resolved. Oct Proposed Technical Report.	Jan Working Documents. July Preliminary dpANS.
1985		Jan Forward Technical Report to SC5.	Jan dpANS forward to X3. July Public Review & Comments.
1986			Jan Forward to ANSI Board.

表1. GKS-3D および PHIGS 作業予定

ISOにおいて、将来検討するように働きかけた。

WG2会議における長い議論の末、以下のような方針で作業を進めることが決定された。⁶⁾

- (1) オランダは、別記{(2-1)参照}のような基本方針の元、GKSの“simple 3-D extension”をTechnical Reportにまとめる。将来のGKS改訂の際には、Technical Reportの内容はGKS本文に吸収されるものとする。
- (2) WG2は、別記{(3-1)参照}のような基本方針の元、アメリカ(ANSI)が新たなWork itemを提案するように、推薦する。PHIGSはその作業の基礎として適当である。

以上が、GKS 3次元拡張に関する現在までの動きであり、今後表1に示したような予定で作業が進行するものと期待されている。

[2]にやや詳しく述べるように、(1)の“simple 3-D extension”(GKS-3D)はGKSとの適合性を完全に保持しながら、各国の意見を取り入れた折衷案となっている。Viewing pipelineの観点から見ると、西独案の2D-3D parallel pipeline、元のオランダ案の3D-2D serial pipelineに対して、新たに提案されたGKS-3Dは、IDIGSやPHIGSと同じく(2D-3D) integral pipelineとなっている。

[2] GKS-3D⁷⁾

オランダが中心となって原案を作成しつつあるGKSの“simple 3-D extension”(GKS-3D)の概要を記す。以下の内容は最終案ではなく、今後の議論により改訂されていくものである点に注意されたい。内容の不明確な点、また誤りと思われる点もあるが、一応原文書に忠実に紹介する。

(2-1) 概要

GKS-3Dは次のような機能を提供する。

- a. 3次元グラフィック基本要素の定義と表現。
- b. Viewing transformation の扱い。
- c. 隠線、隠面消去をも含む表現の扱い。
- d. 3次元入力の扱い。

この拡張は、viewing transformationなどをハードウェアで処理することも含めて、グラフィック装置の有効な利用を可能とする。

GKS-3Dは次のような原則に従って設計される。

- a. 従来のGKSフォーマットは変更なしに走ること。
- b. 可能な限り、3次元機能は2次元機能の一般化とすること。
- c. 2次元、3次元機能の関係は明確に規定されていること。
- d. 既存の2次元機能を変更しないこと。
- e. 3次元拡張のため以外に、新しい機能を導入しないこと。

(2-2) GKS との関係

GKS-3Dシステムにおいて、従来のGKSプログラムが変更なしに走るために、次のように考える。2次元座標 (x, y) に対してZ座標を付加して、 $z=0$ の値を与えて、見かけ上3次元座標に変換する。GKS-3Dのviewing pipelineは後出の図1のようになっているが、normalization transformation や workstation transformation に対しては、同様に適当なZ座標成分を与えて3次元化する。また、viewing transformation としては、Z軸方向のparallel projection を規定する。二つのようにして、従来の2次元GKSのグラフィック要素は、新たな3次元GKSグラフィック要素と混在した状態で、3次元viewing pipelineを通過し、従来と同じ効果を得ることが出来る。

以下に、3次元拡張機能を概説する。

(2-3) Graphical Output

GKS-3Dには、polyline³, polymarker³, fill area^(set³), text³, cell array³, generalized drawing primitive³ (GDP³)なる6種のoutput primitivesがあり、GKSにおける対応するoutput primitivesの、3次元への自然な拡張とされている。Fill area^(set³), cell array³ および text³は、空間内の任意の平面上で規定されるplanar primitivesである。Fill area set³の1つ々々である点列の平面性判定はimplementの仕方に依存している。Planar primitivesには表と裏の区別がある。たとえば、fill area set³においては、用いた面分を規定する点列が時計方向に並んでいるように見える方が表側である。表側が見える、とするのが普通であるが、表裏の見え方は一般に隠線、隠面消去法に依存している。

GKS-3Dのoutput primitive attributesのうち、non-geometric attributesと

<u>POLYLINE3</u>	'number of points	'points		
<u>POLYMARKER3</u>	'number of points	'points		
<u>TEXT3</u>	'text position	'character string		
<u>FILL AREA SET3</u>	'number of points	'points		
<u>CELL ARRAY3</u>	'cell parallelogram	'dimension of colour index array	'colour index array	
<u>GENERALIZED DRAWING PRIMITIVE3 (GDP3)</u>	'number of points	'points	'GDP3 identifier	'GDP data record

<u>SET CHARACTER UP VECTOR3</u>	'character up vector3
<u>SET CHARACTER PLANE VECTOR3</u>	'character plane vector3
<u>SET PATTERN REFERENCE POINT3</u>	'pattern reference point3
<u>SET PATTERN UP VECTOR3</u>	'pattern up vector3
<u>SET PATTERN PLANE VECTOR3</u>	'pattern plane vector3

identification attributes は GKS にあけるものと同様である。Geometric attributes は 3次元を規定するために拡張される。すなわち、text の棄っている平面と text の方向を規定するために、

character up vector と character plane vector の 2つの vector が必要である。また fill area set において、面分内を埋める pattern を規定するために、基準位置を指定する pattern reference point, pattern の棄っている平面と pattern の方向を規定するための pattern up vector と pattern plane vector が必要となる。これらのデータを元、適当な幾何学的変換が施されて、pattern は fill area 内に埋め込まれる。

GKS-3D の新しい attribute として view index がある。これは (2-4) で説明する viewing pipeline における viewing transformation を選択するためのものである。Viewing transformation は workstation 依存であり、各々の workstation には view bundle table があり、view index はこの bundle table に対する index となる。全ての output primitives は、生成時に view index と結合される。

(2-4) Coordinate Transformation

GKS-3D においては、world coordinate (WC3), normalized device coordinate (NDC3), normalized projection coordinate (NPC),

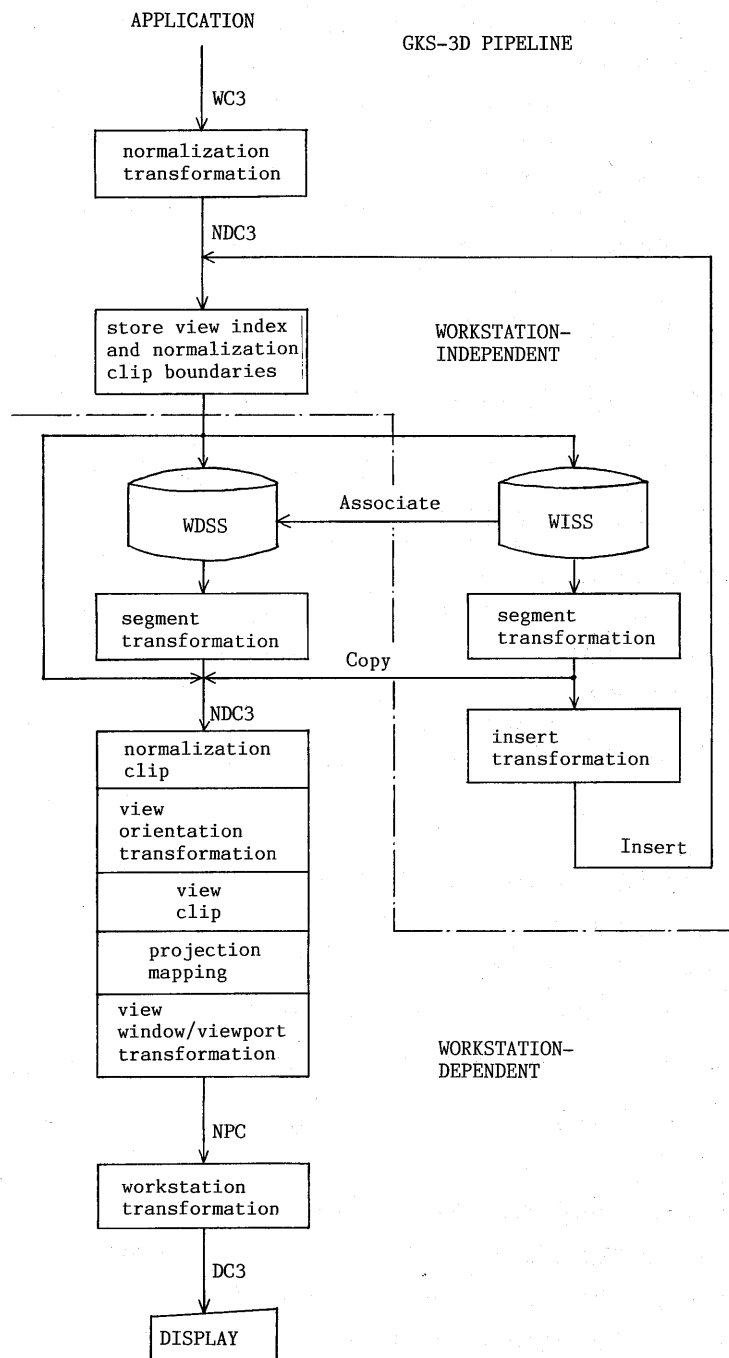


図1. GKS-3D の viewing pipeline

```

SET WINDOW3 'transformation number 'window3 limits
SET VIEWPORT3 'transformation number 'viewport3 limits
SET VIEW INDEX 'view index
SET VIEW REFERENCE POINT3 'view reference point3
SET VIEW UP VECTOR3 'view up vector3
SET VIEW PLANE NORMAL3 'view plane normal3
SET VIEW 'workstation identifier 'view index 'coordinate switch 'error indicator
SET VIEW CHARACTERISTICS 'workstation identifier 'view index 'window clipping
indicator 'far clipping indicator 'near clipping
indicator
SET PROJECTION MAPPING 'workstation identifier 'view index 'projection type
'projection reference point/projection direction
'view window limits 'view plane distance 'near distance
'far distance 'projection viewport limits
SET HIDDEN LINE/SURFACE METHOD 'workstation identifier 'hidden line/surface method
'view index
SET HIDDEN LINE/SURFACE STYLE 'workstation identifier 'hidden line/surface style
SET WORKSTATION WINDOW3 'workstation identifier 'workstation window3
SET WORKSTATION VIEWPORT3 'workstation identifier 'workstation viewport3

```

および device coordinate (DC3) が存在し、全て 3 次元座標系である。図 1 に全体の座標変換の流れを示す。Normalization transformation および workstation transformation は、3 次元変換ではあるが、概念的には GKS の場合と同じである。

3 次元に特有の viewing transformation は大略、次の 3 段階で実行される。

- (i) View orientation transformation: NDC3 から、projection mapping に都合の良いいわゆる UVN 系へ変換する。UVN 系の原点は view reference point、V 軸は view up vector3、N 軸は view plane normal3 にとりよえられる。
- (ii) Projection mapping: 指定に応じて parallel/perspective projection を行い、view volume を設定する。まず、UVN 系の原点から view plane distance だけ離れた UV 面を view plane とする。View window limit に従って view plane 上に view window を設定し、parallel か perspective かにより、projection reference point および projection direction を用いて view window を底とする 4 角錐または 4 角柱をつくる。Near distance および far distance に設定された UV 面でこれを切り取り、角錐台または平行 6 面体を構成する。これが view volume である。更に、shearing function (と perspective transformation) を施して、view volume を直方体に変換する。
- (iii) View window/viewport transformation: view volume を NPC における単位立方体内に設定された projection viewport へ変換する。

上記の過程で、指定に応じて normalization clip および view volume clip がある。隠線、隠面消去は、NPC 系においてなされ、方法やスタイルを指定できるが、これは implementation に依存している。

(2-5) Input

3次元に特有の入力は locator 3 と stroke 3 である。Logical input device の基本概念は2次元の場合と同様であるが、利用可能な入力装置に応じて色々な問題が発生してくる。通常の入力装置は2次元である。2次元入力装置から3次元データを入力する方法は各種考えられるが、未だ詳細に検討されていない。

<u>INITIALIZE LOCATOR3</u>	'workstation identifier 'locator device number 'initial locator position 'initial normalization transformation number 'prompt and echo type 'echo area 'locator data record
<u>SET LOCATOR3 MODE</u>	'workstation identifier 'locator device number 'operating mode 'echo switch
<u>REQUEST LOCATOR3</u>	'workstation identifier 'locator device number °status °normalization transformation number °locator position
<u>SAMPLE LOCATOR3</u>	'workstation identifier 'locator device number °normalization transformation number °locator position
<u>GET LOCATOR3</u>	°normalization transformation number °locator position

(2-6) その他の拡張機能

<u>INSERT SEGMENT3</u>	'segment name 'transformation matrix
<u>SET SEGMENT TRANSFORMATION3</u>	'segment name 'transformation matrix

<u>SCALE3</u>	'scale factor vector °transformation matrix
<u>TRANSLATE3</u>	'translation vector °transformation matrix
<u>ROTATE X</u>	'rotation angle °transformation matrix
<u>ROTATE Y</u>	'rotation angle °transformation matrix
<u>ROTATE Z</u>	'rotation angle °transformation matrix
<u>ACCUMULATE MATRIX3</u>	'transformation matrix A 'transformation matrix B °composed transformation matrix
<u>TRANSFORM POINT3</u>	'point 'transformation matrix °transformed point

[3] PHIGS^{5,9)}

(3-1) 概要

PHIGS は Programmer's Hierarchical Interactive Graphics Standard の略で、ANSI X3H31 において 表1 のような予定で作業が進められている。これは、次のような要求に応えることを目的としている。

- 高度のインターアクション
- グラフィックス・データの階層的構造化

- グラフィック・データやデータ間の関係の変更
- 幾何学的変換の一般的扱い
- 3次元グラフィック・データの扱い

特に、グラフィック・データの変更、幾何学的な関連、これらの高速度動的取扱い、を重視している。

応用分野として、CAD/CAM、ロボセ又制御など多くのものがあげられている。PHIGSを用いることの利点として、特に、従来の応用プログラム側で行なわれていた機能がPHIGSによって遂行されるため、応用開発に専念できるようになる、ということをおげている。

PHIGSは、GKSとはその constituency が異なることを主張しているが、それでもGKSと最大限の適合性を保つべく、次のような原則を定めている。

- GKSとPHIGSとの間の技術的相関は、それがPHIGSの constituency および目的から必要な時にみ許される。
- GKSとPHIGSが同じ機能を持つている時は、そのパラメータや言語結合方式などは共通のものとする。
- GKSを越える機能については、可能な限り、GKSの機能を default とし、て含むようにする。

(3-2) 機能概要

(i) 全体構成

PHIGSの全体構成はGKSとよく似ており、workstationの概念、bundle tableの扱い、deferred stateの制御など、ほぼ同様の機能を有している。

(ii) Graphics data content

PHIGSにはpolyline, polymarker, text, fill area, polygon set, cell array, cell run array, GDPの8種のoutput primitivesがあり、2D/3D両者が用意されている。これらのoutput primitivesが使用されると、後述するstructureに保持され、structure traversalにより出力されることになる。

output attributeの扱いもGKSとほぼ同様で、individual/bundle settingがある。

(iii) Transformations

viewing pipeline を図2に示す。Modeling transformationがPHIGSに特有のもので、structureと関係づけられ、階層構造の上位から下位へとたどるなか適切なmodeling transformationを合成するような仕組みが考えられている。Viewing transformationやworkstation transformationはGKS-3Dとほぼ同様のもののようである。

(iv) Graphic data organization

PHIGSのグラフィック・データは階層的構造を持つstructureとして管理されている。Structureのデータ要素は、output primitives, attribute settings, transformations あるいは他のstructureの参照から成っている。Structureはその上位のstructure

の attribute を受け継いでいる。

Structure は各 workstation
へ post された時に、始めて表
示される。表示のためにその
structure を traverse する必要
があり、他の structure の参照
も含めて、attribute binding など
の方式が決めている。

表示されている picture の
修正のための豊富な制御機能
が、structure を用いて実現さ
れている。あたかも通常の
text editor を用いるかごとく、
structure の任意の要素を挿入、
削除、変更するなどができる。
その他、structure 全体の edit
や archiving の機能もある。動
的な structure の変更に対して、
workstation の deferred/implicit
regeneration mode に従って、実
際の表示が変更される。

(V) Graphics input

GKS と同様な logical input
class (locator, stroke, valuator,
choice, pick, string) と mode
(request, sample, event) が用意
されている。Pick 入力に対し
ては、structure の traversal
sequence をキーとして返す。

(3-3) 推

PHIGS の全 function 数は 190 ほどもあり、文書も 330 頁余りの大部なもので
ある。今後更に増強されるかもしれない。Test implement も進められているよう
であるが、仕様の面でも implement の面でも多くの未検討の問題が残っているように
も思われる。

参考文献

- [1] GKS V9.2, ISO/TC97/SC5/WG2 DIS9442 (1982)
- [2] IDIGS V4.0, ISO/TC97/SC5/WG2 N151 (1982.5)
- [3] 3D Module for GKS, DIN-AK5.9.2 11-83 (1983.9)
- [4] GKS Extension in 3D, NNI (1983.9)
- [5] PHIGS baseline document, X3H31/82-03R03 (1983.11)
- [6] Report of 3D subgroup, ISO/TC97/SC5/WG2 N193 (1983.9)
- [7] GKS-3D (draft), NNI (1983.12)
- [8] J. Foley & A. van Dam, Fundamentals of Interactive Computer Graphics, Addison-Wesley (1982)
- [9] PHIGS, SD-3, ANSI X3H3/83-89R1 (1983)

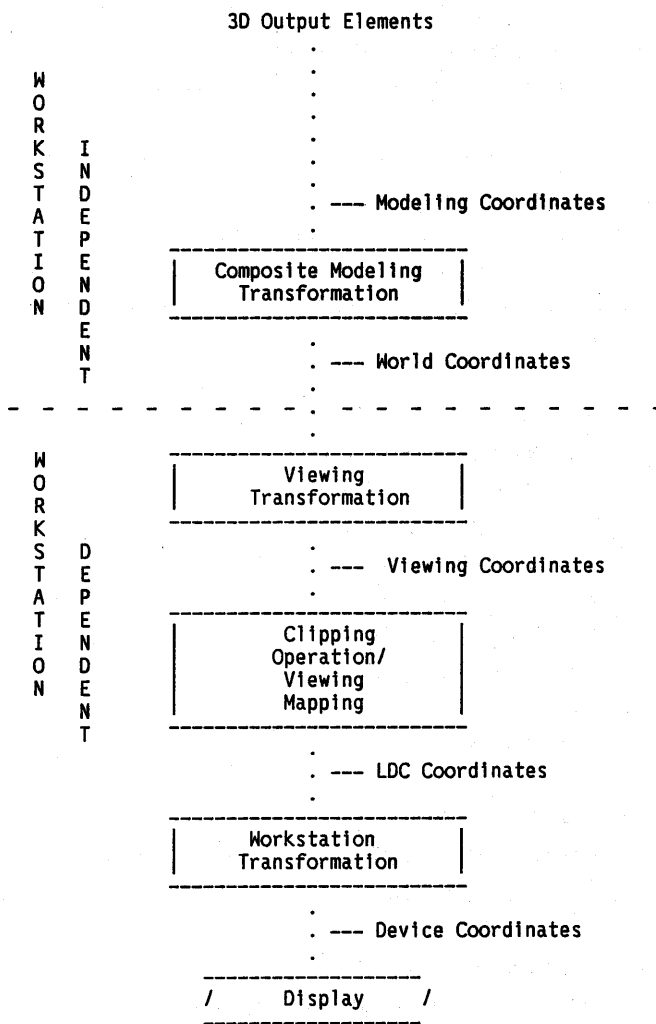


図2 PHIGS の viewing pipeline