Computer Graphics Standards

Miente Bakker
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Already for more than one decade there is a strong interest at CWI in computer graphics standards. Since the pioneering days in this area, we have contributed to the development of these standards and have engineered implementations, drivers for advanced graphical workstations, applications on top of standards and related software. In this article, a survey of CWI activities concerning graphics standards is given. The article consists of the following parts: 1. Introduction; 2. Outline of a Computer Graphics Reference Model; 3. Interest of the Interactive Systems Department in Computer Graphics Standards; 4. Future developments. Appendix on Non-Uniform Rational B-splines (NURBS).

1. Introduction

1.1. International standards on computer graphics

The International Organisation for Standards (ISO) publishes standards on several fields of science and technology where there is a need for normalisation. Especially in the field of Information Technology, there is an increasing flow of new standards. Examples of such standards are the Open System Interconnection (OSI) standards, the office automation standards and the standards for programming languages like ADA, FORTRAN 77, PASCAL and ALGOL 68.

All these standards are produced by international teams of experts, after they have reached consensus on their contents at international meetings. These teams consist of representatives from national standard organisations like the American National Standard Institute (ANSI), the British Standard Institute (BSI) and the Deutsches Institut für Normierung (DIN). The Netherlands is represented by the Nederlands Normalisatie Instituut (NNI).

Since 1985, there also exist international computer graphics standards, three of which have played and still play an important role in CWI activities: GKS [1], GKS-3D [2] and PHIGS [3]. All these standards are produced by the several working groups of ISO SC 24, the subcommittee of ISO that is responsible for standards on computer graphics. See Arnold & Bono [10] for a more detailed description of the production process of international standards on computer graphics.

CWI Quarterly 3, 237-253

The making of GKS

In the late 1970s the need for graphics standards became pressing, because there were several de facto standards (e.g. CALCOMP, DISSPLA and GINO) which were not generally available and were incompatible with each other.

The first international conference on a computer graphics standard was organised by IFIP at Seillac, France in 1976 [24]. Nine(!) years later, the first international standard for two-dimensional computer graphics was published: GKS, the Graphical Kernel System. In these nine years, at least ten international meetings were spent on the development of GKS. Reaching consensus on GKS was very time-consuming for several reasons:

- lack of experience at the beginning; the experts had to start from scratch, because there were no standards that could serve as a model;
- competing national interests, e.g. the ANSI interest in the national 3D graphics standard CORE; because several US vendors of graphical workstations had manufactured CORE machines, ANSI was very reluctant to accept GKS, to say the least.

For a more detailed description and chronology, see Enderle et al. [15].

The success of GKS

GKS was not an immediate success. The early implementations were generally poor and established de facto standards like CORE [11] and GINO did not give way easily. However, it gradually gained ground and is now *the* standard for 2D computer graphics. Its market is still growing, among other things because several European, Japanese and US government agencies are obliged to use GKS.

1.2. Current computer graphics standards

The very first International Standard on Computer Graphics is the Graphical Kernel System (GKS).

Once GKS had been produced and a high level of experience in developing standards had been acquired within ISO, other standards soon followed:

- The Computer Graphics Metafile (CGM) [4], a standard for long-term storage and transfer of graphical data; this standard is becoming very popular, because it can serve as a system-independent link between different graphics systems; this makes it very valuable for the communication between graphics systems over networks;
- GKS-3D [2], a three-dimensional extension of GKS; the future of this standard is not yet clear; its market will certainly not be as big as the GKS market;
- PHIGS [3] (the Programmer's Hierarchical Interactive Graphics System), a 3D computer graphics standard, which strongly resembles GKS-3D, but which has a more advanced model for the storage, manipulation and archiving of data; its market has yet to be developed, among other things

because the development of its extension PHIGS PLUS [7] has not yet been completed;

- several specifications of GKS, GKS-3D and PHIGS in programming languages like FORTRAN 77 and ADA were published (the semantics of standards like GKS and PHIGS is independent of existing programming languages, so for such standards, a separate specification in a programming language, a so-called language binding is needed); examples are GKS FORTRAN, PHIGS ADA.

2. Description of computer graphics standards by means of a reference model

Now that a family of graphics standards has been produced, an attempt is being made by computer graphics experts in ISO to develop a so-called Computer Graphics Reference Model [6]. This is an abstract graphics system that could serve as a model for the next generation of graphics standards. The general feeling among graphics experts in ISO is that the series of graphics standards produced in the period 1985-1989 is to be considered as the *first generation* of standards and that the next generation should be based on the Reference Model. In developing this Reference Model, the know-how and experience acquired during the production of standards in the past ten years and the current state of the art in computer graphics technology are being used.

The current Reference Model is not yet stable and it is expected to take at least two years, before it stabilizes. In the next few sections, we give a brief outline of it, which may be subject to modification.

2.1. Major concepts used in the Computer Graphics Reference Model

The following concepts are being used in the reference model

- Output; examples are:
  - POLYLINE; a sequence of connected straight lines;
  - POLYMARKER; a set of glyphs centered around positions;
  - TEXT; a string of characters;
  - POLYGON FILL; the filling of the interior of a polygon, e.g. with a hatch, a colour, or a pattern.

  Associated with these *output primitives* are several output properties or attributes, like linewidth, marker colour, text font, polygon fill style;
- Projection Transformations;
  e.g. rotation, scaling, clipping, perspective projection;
- Input;
  the input tokens brought in by the operator through an input device (e.g. mouse, track ball, key board) are partitioned into some classes; the most important ones are:
  - CHOICE; input is selected from a finite number of alternatives, e.g. a pop-up/pull-down menu, or from a number of buttons;
  - VALUATOR; input is selected from a range of real values, as represented by e.g. a dial;

Importance of international standards

For several reasons, a computer graphics standard is important. A short overview is given here.

- Portability, independence of configuration;
  the availability of GKS on any computer figuration guarantees that application software running on GKS can easily be ported from one installation to the other;
- Independence of user;
  if the user (e.g. a computing centre, the CAD department of an industry) is discontent with the performance of a GKS or PHIGS implementation, the implementation can be replaced by another implementation, because the vendor only has copyright on the implementation, not on the specification;
  Communication of graphical data;
  the availability of a Computer Graphics Metafile (CGM) interpreter and generator on any computer graphics environment guarantees that GKS- or PHIGS-produced metafiles can easily be interpreted on any other graphics environment;
- Common vocabulary;
  the development of graphics standards has led to a de facto standardisation of definitions and methodology. To-day concepts like workstation, polyline, viewport, choice device and metafile are commonplace in computer graphics.

  - STRING; a text is typed on a keyboard or selected from a window by a mouse;
  - LOCATOR; a single point on the display screen sent in by the operator;
  - STROKE; a sequence of points on the display screen sent in by the operator;
  - PICK; (reference to) a part of the picture on the display is returned to the application program.
- Picture;
  a composition of output primitives with their properties, e.g. red thick polyline + green polymarker + text;
- Collections;
  grouping of a set of graphical entities (e.g. output, output properties, transformations); examples are segments in GKS or structures in PHIGS;
- Data Capture;
  a mechanism for representing a picture for storage, retrieval and transmission; an example is the picture capture in the Computer Graphics Metafile.

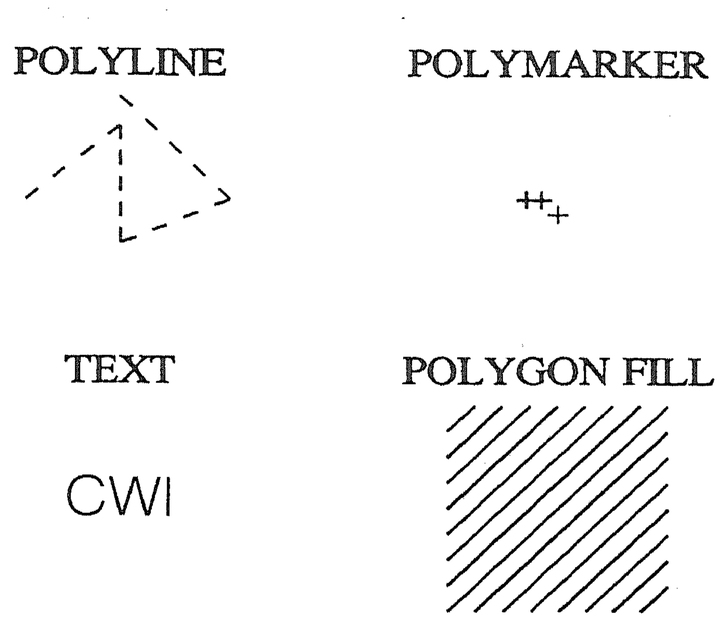

FIGURE 1. Computer graphics output primitives

2.2. Subdivision of a graphics system in environments

Within the current reference model, a graphics system is partitioned in five environments or layers:

- The application environment;
  in this layer, the output data enter the graphics pipeline or the input data are given to the application; the output or returned input are given in User Coordinates, Master Coordinates, Modelling Coordinates or World Coordinates, depending on the standard;
- The virtual environment;
  before output data enter the viewing pipeline within a workstation, they are mapped to an abstract picture, which is defined in a so-called virtual coordinate space; this mapping is necessary to transform data from different coordinate spaces to data in one single space, thus composing a virtual picture; here the data are given in Virtual Device Coordinates or Normalized Device Coordinates, depending on the standard;
- The projection environment;
  in this layer, a specific view at the scene is taken which produces the picture to be presented;
- The logical environment;
  in this layer, the picture is stored in the logical workstation, before it is displayed;
- The physical environment;
  in this layer, the picture is displayed on the screen or the input data are captured by an input device.

The application and virtual environment are device independent, the other three environments are device dependent.

3. Interest of CWI in Computer Graphics Standards

From the mid-1970s (if we ignore the X1 plotter), CWI has had a strong interest in Computer Graphics. In this section we describe the main features of CWI's involvement.

3.1. Early activities; development of GKS

From the very beginning, CWI participated actively in the development of Graphics Standards. This was demonstrated in the following activities:

- the ILP project;
  from 1976-1978, a CWI working group has developed a graphics system ILP [12] (Intermediate Language for Pictures), which shows much resemblance to the later international standard PHIGS;
- Participation in ISO meetings developing GKS;
  Almost from the beginning, CWI people have attended the numerous ISO meetings where GKS was developed. Paul ten Hagen has been chairman (1977-1983) of the working group that was responsible for GKS;
- Pilot implementation of GKS in C;
  Together with David Rosenthal from Edinburgh University who worked at CWI in 1982, CWI researchers, most notably Bèhr de Ruiter, developed one of the first implementations in C of GKS; when it became clear in 1982 that GKS would become an international standard, this implementation was among the few ones available.

3.2. The CWI C implementation of GKS

When GKS broke through as an International Standard in 1982 (despite stiff opposition from market-leading American companies united in ANSI), there were almost no implementations available, because the American software industry had backed the CORE [11] standard. The C implementation of CWI was among the few available ones. This circumstance has created a demand for this product, which is still growing. The following CWI activities were done to satisfy this demand:

- Documentation;
  On request of a large international manufacturer of graphical workstations, a 2 volume Reference Manual of GKS [13] (some 600 pages) and a 200 page GKS User Guide [14] were produced.
  With assistance of some experts from the Dutch universities, this project was successfully carried out, in spite of a bumpy start (the original document editor misinterpreted his task and copied entire paragraphs verbatim from the ISO GKS document [1]; after the customer balked, the work of this document editor was taken over by others);
- Device drivers;
  For several advanced graphical workstations (Sigmex, Laser Printers with PostScript [16] interpreter, IBM 5080, SUN 3), device drivers were developed, sometimes on request of workstation vendors;
- FORTRAN shell around GKS/C;
  around the C implementation a FORTRAN shell was made; this approach

proved to be much more successful than a full FORTRAN implementation (see chart on page 245);

- Certification of GKS/C;
  GKS/C and its FORTRAN shell was certified for some workstations by a test suite from the German Gesellschaft für Mathematik und Datenverarbeitung (GMD) ; GMD had developed a collection of GKS FORTRAN test programs to test a GKS FORTRAN implementation on several conformance aspects:
  - correctness of modules; does every module do what it is supposed to do? e.g. if SET POLYLINE INDEX (3) has been called, will INQUIRE POLYLINE INDEX give the value 3?
  - operator tests;
    do the operator test programs produce similar pictures as in the test suite manual?
  - error handling;
    if a GKS function is called under wrong conditions, will the implementation produce the correct error message? after the implementation had been thoroughly tested, it got a certificate from GMD, in spite of the terrible shortcoming that the backslash symbol ('\') could not be reproduced;
- Enhancements;
  Some additional features were added to GKS/C which enabled among others panning and zooming of pictures [18]; this was realized by the introduction of the so-called segment grouping.

3.3. GKS-3D

3.3.1. The standard. In its early days of development, GKS was intended to be a standard for both 2D and 3D. Because the development of GKS took so much time, the 3D part was removed from GKS and its development was delayed until after the completion of GKS.

When GKS was functionally complete in 1982, the developers turned to the development of GKS-3D. In order to guarantee a rapid production of GKS-3D, some limitations were posed on it:

- Upward compatibility with GKS; any application program running on GKS should also run on GKS-3D;
- 3D input and output are strictly limited to 3D extension of existing GKS functionality;
- Hidden Line Hidden Surface Removal (HLHSR), an essential part of 3D graphics, must be defined as vaguely as possible, in order not to impose too strict limitations on the development of HLHSR algorithms;
- The 3D viewing pipeline must be compatible with the 2D viewing pipeline.

Especially the last condition proved to be very problematic. The main cause was that Normalization Transformation, which had been very useful in 2D GKS, was less suitable for 3D viewing. Nevertheless, a viewing pipeline [20] was designed for GKS-3D, which was also adopted by PHIGS.

3.3.2. Implementations. Parallel to the design of GKS-3D, a C-implementation of it was made by CWI, together with a FORTRAN shell. This project was completed successfully, but the product has not yet been marketed, because the market for 3D graphics so far has been dominated by PHIGS.

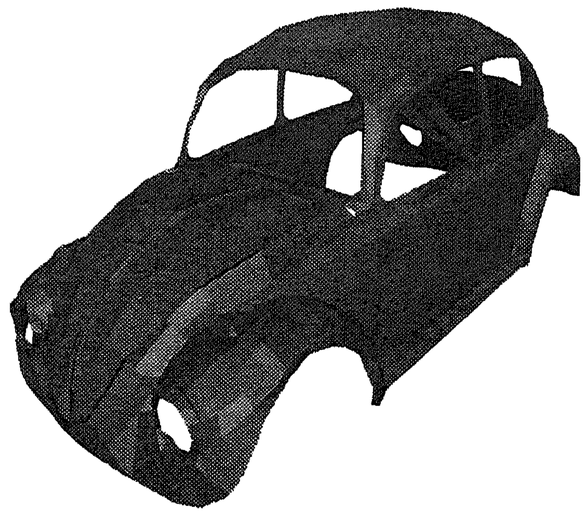

FIGURE 2. Car body drawn by GKS-3D

4. FUTURE DEVELOPMENTS IN COMPUTER GRAPHICS STANDARDS

4.1. GKS revision

Every five years, an International Standard has to be revised. The revision of GKS is due in 1990. From 1987, however, several revision activities have been developed, in and outside ISO. It started with a Eurographics workshop in Disley [19] and was continued at several ISO meetings.

The main topics of the GKS revision were:

- the improvement of the input and the text model, based on years of GKS experience;
- addressing technology that did not yet exist in 1982, like windowing, raster graphics, networking;
- improvement of the data storage mechanism;
- correction of errors like editorial errors, technical errors, deprecated (ISO terminology for obsolete, ill-defined) functionality.

Since March 1989, a project group of ISO SC 24 has done some work on the GKS revision. An initial draft [9] was produced in August 1989. This draft, however, was rejected by most member bodies of ISO SC 24, because it was supposed to widely exceed the project specification.

GKS C/FORTRAN Interface

The UNIX system allows C modules to be called from FORTRAN application programs, provided that the function name satisfies some conventions. Thus, the GKS FORTRAN subroutine

```
SUBROUTINE GPL(N, X, Y)
INTEGER N
REAL X(N), Y(N)
<body>
END
```

can be simulated in C by the function

```
int gpl_(n, x, y)
int *n;
float *x, *y;
{
<mapping of *x and *y to data types in C GKS;
then call of function polyline( ) >
}
```

This method of simulating a FORTRAN implementation (it can also be applied to other languages like PASCAL) has several advantages:

- allocation of storage, an unknown feature of FORTRAN 77, can be done without problems, because C does support allocation;
- the full richness of the C language can be exploited.

4.2. Emerging new standards

Currently, some new ISO standards on computer graphics become of interest. We briefly describe them in this section.

4.2.1. New Application Programmer's Interface. Besides the GKS revision project, another project is going on: the New Application Programmer's Interface project (new API). Some ISO members think that the revised GKS may be obsolete very soon, and are studying the possibilities of an entirely new computer graphics standard that is to succeed both GKS and PHIGS. A draft has not yet been made, although some ISO SC 24 members expect that the proposal for the GKS revision [9] (see also 4.1) might be a good starting point.

4.2.2. Other standards. Some other computer graphics standards are emerging as well. We briefly summarize them:

- Computer Graphics Interface [5] (CGI);

a six part standard describing the interface between a client program (GKS, PHIGS) and an arbitrary workstation, like CGM; CGI is strongly compatible with CGM; almost every graphical entity of CGM is supported by CGI; it is also strongly oriented towards *X*-windows [17];
CGI is expected to be published in 1991;

- Several supplements to CGM;
  1) addition of 3D functionality to CGM to make it suitable as a GKS-3D Metafile;
  2) addition of advanced 2D functionality to CGM, such as Bézier splines, line caps, line joins, etc.;
- Image Processing and Interchange (IPI);
  a new project on image processing based upon the German Iconic Kernel System and the American PIK Strawman; publication is expected in 1993-1994;
- Several language bindings and encodings of CGI, PHIGS, GKS-3D.

4.3.2. PHIGS PLUS. This is an extension of the graphics standard PHIGS [3]. PHIGS PLUS [7] (an acronym for PHIGS Plus Lumière Und Surfaces) is being developed, because several areas in technical engineering (CAD, CAE, Scientific Computing) need more advanced visualization than GKS or PHIGS can support. PHIGS PLUS extends PHIGS into three directions:

- Curved lines and surfaces;
  Whereas GKS and PHIGS only support the drawing of piecewise straight lines and piecewise flat surfaces, PHIGS PLUS supports the drawing of curved lines and surfaces.
  For this purpose, the designers have selected one of the most popular classes of blending functions in geometric modelling: the Non-Uniform Rational B-Splines (NURBS) [23, 25]. See the Appendix for a description of the NURBS;
- Introduction of *direct colour*: GKS and PHIGS only support indexed colour representation; PHIGS PLUS offers the possibility to specify colours directly;
- Primitives with data;
  GKS and PHIGS only support primitives that are represented by *control points* on one hand (polyline, polymarker, polygon filling) and on the other hand by a list of *primitive properties* (the line style, the marker type, the fill colour). These two entities (the control points and the properties) are separate.
  PHIGS PLUS supports the concept of output properties associated (directly or indirectly) with the control points of the primitive.
  Examples of such primitives are:
  - POLYLINE with colours associated with the vertices; this property makes it possible to draw POLYLINES with internally variable colours;
  - FILL AREA SET WITH DATA; several kinds of data can be associated with the vertices or the facets, like VERTEX COLOUR, VERTEX NORMAL, FACET COLOUR, FACET NORMAL.
  - TRIANGULAR SET WITH DATA; this primitive is also extremely useful for Finite Element applications, because it corresponds with linear triangular splines [26];

Nice properties of splines

Splines have several properties which make them highly popular in several areas of technology, like scientific computing, Computer Aided Geometric Design [26], and Computer Graphics [23]. The main properties are:

- Local control;
  if one of the control points of a spline curve or surface is shifted, the curve does not change overall, but only in the direct environment of the control point shifted (see Figure 3); this property is due to the fact that each curve segment depends on only a few adjacent control points; e.g. each segment of a cubic NURBS curve depends on 4 consecutive control points; this property enables local correction/update by shifting single control points;
- Parallel rendering of curve/surface patches;
  A spline curve/surface consists of a number of segments (patches), each of which depends on only a few control points; this property enables efficient parallel rendering of the curve/surface;
- Scientific Computing;
  Several problems in technical engineering (heat conduction, mass transport, optimal control, curve fitting) can be modelled in terms of NURBS or splines that can be mapped to NURBS; this often leads to large linear systems with *sparse* matrices; the reliability and the numerical stability of this approximation method makes splines highly popular with numerical analysts;
- Modelling of quadric curves and surfaces;
  This class of curves can be represented by rational splines in a computationally very attractive way. Thus a half circle can be modelled by

$$x = (1-t^2)/(1+t^2);\ y = 2t/(1+t^2);\ -1 \leqslant t \leqslant 1.$$

  which is much more CPU-efficient than

$$x = \cos(t);\ y = \sin(t);\ -\frac{\pi}{2} \leqslant t \leqslant \frac{\pi}{2}.$$

- Enveloping property;
  a NURBS curve lies within the smallest convex polygon that envelops all the control points;
- Approximation properties;
  Complex curved lines (e.g. sine curves or curves defining text characters) can be approximated by NURBS curves; complex curved surfaces (e.g. car bodies) can be approximated by bivariate NURBS surfaces.

The concept of output properties associated with geometric data is called *multidimensional data.* This kind of data is necessary to support the enhanced rendering capabilities of PHIGS PLUS, like lighting, shading, etc.

- Enhanced rendering, with shading, reflection, lighting, depth cueing (depth effect achieved by linear interpolation between workstation dependent depth cue colour and input colour; interpolation is determined by two depth cue reference planes and two depth cue scale factors, which are maintained on the workstation state list); the rendering attributes that PHIGS PLUS supports are :
  Phong shading, Gouraud shading, ambient reflection, specular reflection, diffuse reflection, transparency.
- Enhanced control over the appearance and invisibility of front and back facing portions of area defining primitives; this is done by the introduction of front and back face and face culling and face distinguishing.

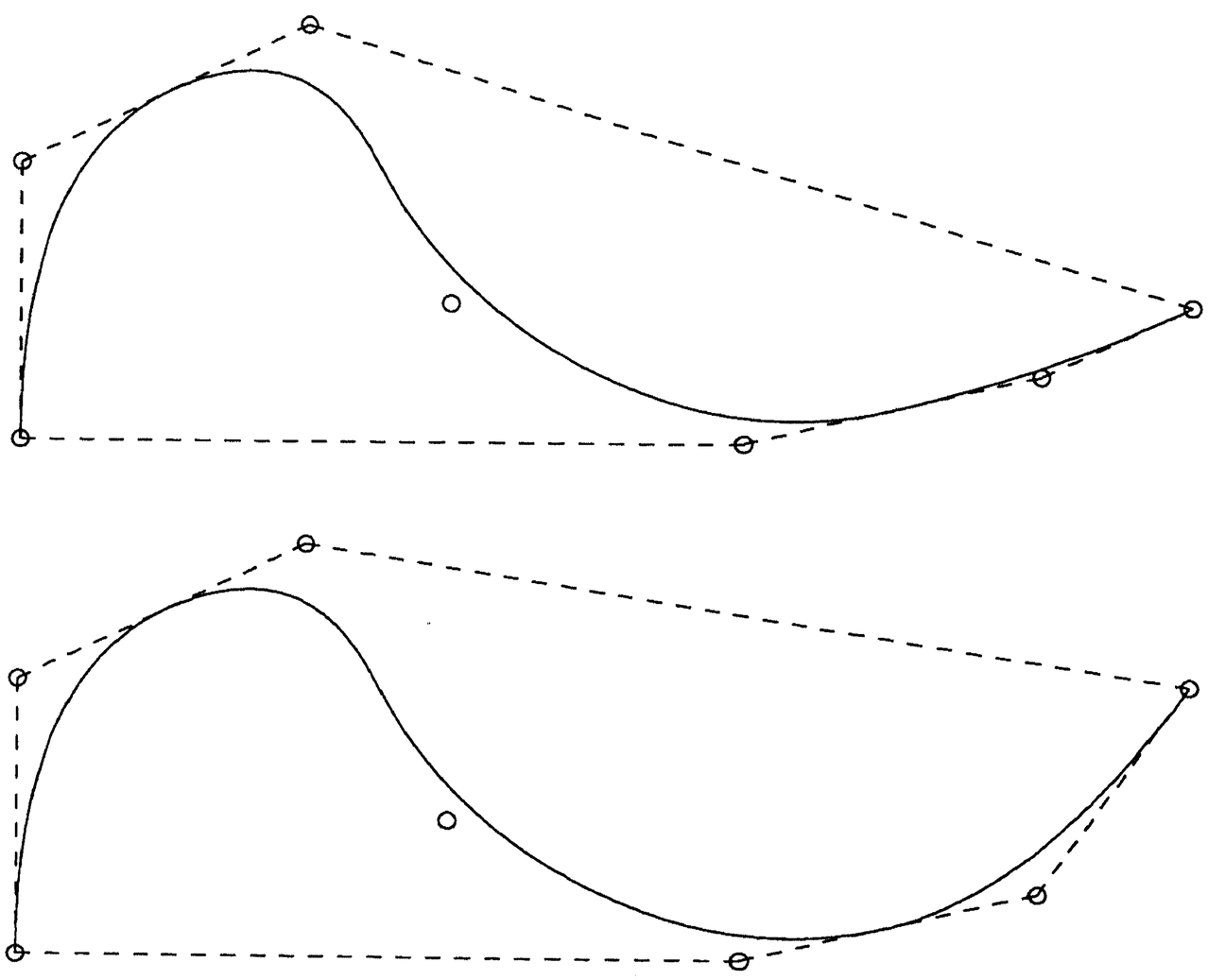

FIGURE 3. NURBS curves with control points (o). In the second curve, the last control point has been shifted. The curve only changes in the direct neighbourhood of that control point. As can be seen, the (dashed) convex hull of the control points completely envelops the NURBS curve.

4.3. CWI interest in a new Application Program Interface

Since the revision of GKS started in 1987, CWI has been interested in revising and modernising it. CWI members have participated in workshops in and outside ISO, where the GKS revision was treated ([19]). A significant contribution to the GKS revision and the new API was given by an improved input model developed by Robert van Liere and some other experts (see [8, 21, 22]).

New challenges to standards

Since the early 1980s, when GKS was designed, life has become more and more difficult for the designers of standards in computer graphics. Some of the new problems are:

- De facto standards;
  Since the mid-80s the ISO graphics standards are increasingly challenged by so-called de facto graphics standards, which are produced by industries or consortia outside the ISO arena; examples of such standards are
  - PostScript [16], a Page Description Language with a very strong graphics component;
  - X11.4 [17], a Window Management System, also with a very strong graphics component;
  - NeWS, another Window Management System strongly relying on PostScript for its graphics component;
  - MAP/TOP, a protocol for office and manufacturing automation;
  - RenderMan [25], a standard for the rendering pipeline of a graphics system;
- Integration with other ISO standards;
  when GKS was functionally complete in 1982, it was the only international standard on graphics; furthermore, there were no standards on other technologies that had to be taken into account (except some standards on technical drawing and character encoding); today there are tens of other standards, that have to be taken into account:
  - the other graphics standards (PHIGS, CGI, CGM);
  - related standards, like the office automation standards (ODA/ODIF, SGML, SPDL), the data base management standards (SQL), the design automation standards (IGES, PDES, STEP), the OSI standards on Open Distributed Processing (ODP), File Transfer Access and Management (FTAM);
- Keeping pace with technology;
  the production process of international standards is painfully slow for several reasons (bureaucracy, competition, geographical problems, large amount of paperwork, etc.); hence a standard, which is already by definition lagging behind the technology it addresses, runs the risk of being completely obsolete even before it is published.

APPENDIX

THE MATHEMATICS OF NON-UNIFORM RATIONAL BASIC SPLINES (NURBS)

Parametric curves are curves of the form

$$\vec{C}(t) = \sum_{i=0}^{N} \vec{P}_i B_i(t)$$

where $\vec{P}_i$ are some control points in the modelling space and where $B_i(t)$ are real-valued functions defined on some interval $[t_{\min}, t_{\max}]$. In fact, a parametric curve involves a complex geometrical mapping from a closed interval of real numbers to a curve in modelling space. These $B_i(t)$ form a basis for a finite-dimensional approximation subspace of a function space. Classical examples of $B_i(t)$ are:

- $B_i(t) = t^i$, $0 \leqslant t \leqslant 1$;
- $B_i(t) = T_i(t) = \cos(i^* \arccos(t))$, $-1 \leqslant t\ 1$; (Chebyshev polynomials);

Today, however, the field of parametric curves is largely dominated by piecewise polynomial curves or spline curves, in particular the basic splines or B-splines. The main reasons for their immense popularity are among others local control, numerical stability, use of parallelism, sparser use of memory.

B-SPLINES

Let

$$t_{\min} = t_0 < t_1 < \cdots < t_{N-1} < t_N = t_{\max}$$

be a non-uniform partition of the interval $[t_0, t_N]$. Then the so-called B-spline functions $B_{i,k}(t)$ are defined as follows:

$$B_{i,0}(t) = \begin{cases} 1, & t_i \leqslant t \leqslant t_{i+1}; \\ 0, & \text{elsewhere;} \end{cases}$$

$$B_{i,k}(t) = \frac{t - t^i}{t_{i+k} - t_i} B_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(t);\ k \geqslant 1;\ i = 0,\ldots,N;$$

Parametric curves of the form

$$\vec{C}(t) = \sum_{i=0}^{N} \vec{P}_i B_{i,k}(t), \quad t_0 \leqslant t \leqslant t_N;$$

are B-spline curves.

These B-spline basis functions have the following properties:

- On each interval $[t_1, t_{i+1}]$, they are k-th degree polynomial; at the knots t_i, they are $k-1$ times differentiable;
- Their values lie between 0 and 1 and they sum to 1:

$$0 \leqslant B_{i,k}(t) \leqslant 1;$$

$$\sum_{i=0}^{N} B_{i,k}(t) \equiv 1;\ t_0 \leqslant t \leqslant t_N.$$

These properties explain the convex hull property (see Fig. 3);

- Each $B_{i,k}(t)$ is identically zero outside the interval $[t_{i-k}, t_{i+1}]$:

$$B_{i,k}(t) \equiv 0,\ t_0 \leqslant t \leqslant t_{i-k} \text{ or } t_{i+1} \leqslant t \leqslant t_N;$$

This property implies that on each interval $[t_i, t_{i+1}]$, only $k+1$ basis functions are non-zero: $B_{i-k+1,k}(t), \ldots, B_{i+1,k}(t)$. For the evaluation of $\vec{C}(t)$, this means that on each interval $[t_i, t_{i+1}]$, $\vec{C}(t)$ depends only on $\vec{P}_{i-k+1}, \ldots, \vec{P}_{i+1}$:

$$\vec{C}(t) = \sum_{i=i-k+1}^{i+1} \vec{P}_i B_{i,k}(t),\ t_i \leqslant t \leqslant t_i;$$

This explains the local control of the B-splines: a change of $\vec{P}_i$ only changes the behaviour of $\vec{C}(t)$ on $[t_{i-1}, t_{i+1}]$.

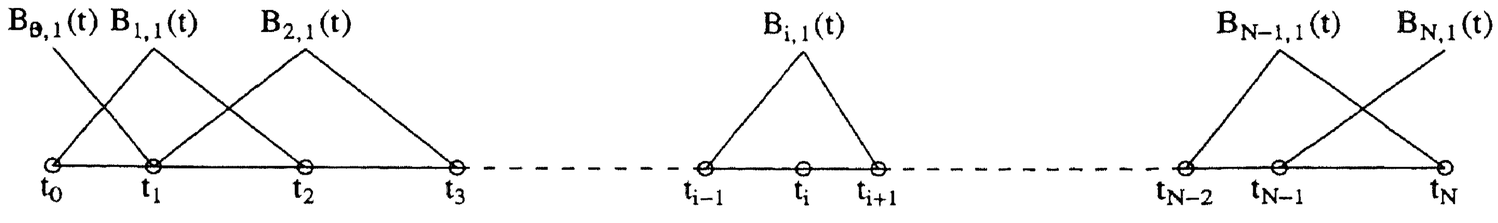

FIGURE 4. Piecewise linear B-spline basis functions $B_{i,1}(t)$

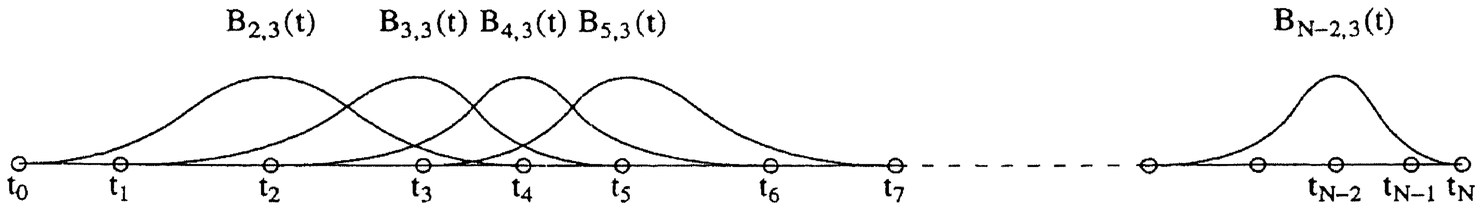

FIGURE 5. Piecewise cubic B-spline basis functions $B_{i,3}(t)$

RATIONAL B-SPLINES

A *Rational* B-spline is defined by the formula

$$\vec{C}(t) = \frac{\sum_{i=0}^{N} \vec{P}_i B_{i,k}(t)}{\sum_{i=0}^{N} w_i B_{i,k}(t)}$$

where $\{w_i\}_{i=0}^{N}$ is a set of *positive* weight factors. Note that a rational B-spline reduces to a normal B-spline, if all the weight factors are identical. These rational B-splines can be used for modelling quadratic curves, like the circle and the ellipse.

RATIONAL B-SPLINES AND HOMOGENEOUS COORDINATES

One can interpret the couples $\{\vec{P}_i; w_i\}$ as *homogeneous coordinates*. This interpretation makes it possible to apply projective geometry analysis to NURBS curves.

B-SPLINE SURFACES

These are surfaces of the form

$$\vec{S}(u,v) = \sum_{i=0}^{M} \sum_{j=0}^{N} \vec{P}_{i,j} B_{i,k}(u) B_{j,l}(v);\ u_0 \leqslant u \leqslant u_M;\ v_0 \leqslant v \leqslant v_N;\ k,\ l \geqslant 1;$$

where $P_{i,j}$ is a rectangular array of control points and where $B_{i,k}(u)$ and $B_{j,l}(v)$ are k-th and l-th degree B-splines on $[u_0,u_M]$ and $[v_0,v_N]$, respectively.

Rational B-spline surfaces are surfaces of the form

$$\frac{\sum_{i=0}^{M} \sum_{j=0}^{N} \vec{P}_{i,j} B_{i,k}(u) B_{j,l}(v)}{\sum_{i=0}^{M} \sum_{j=0}^{N} w_{i,j} B_{i,k}(u) B_{j,l}(v)};\ u_0 \leqslant u \leqslant u_M;\ v_0 \leqslant v \leqslant v_N;\ k,\ l \geqslant 1;$$

where $\{W_{i,j}\}$ is a rectangular array of positive weight factors.
As with rational B-spline curves, the couples

$$\{\vec{P}_{i,j};\ w_{i,j}\}_{i,j=0}^{i=M,j=N}$$

can be interpreted as homogeneous coordinates.

REFERENCES

ISO documents

1. Graphical Kernel System (GKS), ISO 7942, 1985.
2. Graphical Kernel System for Three Dimensions (GKS-3D), ISO 8805, 1988.
3. The Programmer's Hierarchical Interactive Graphics System (PHIGS), ISO 9592 (3 parts), 1989.
4. Computer Graphics Metafile (CGM), ISO 8632 (4 parts), 1987.
5. Computer Graphics Interfacing Techniques for Dialogues with Graphical Devices (CGI), Draft International Standard ISO 9636 (6 parts), 1990.
6. Computer Graphics Reference Model, Committee Draft, ISO/IEC JTC1/SC24 N 512.
7. The Programmer's Hierarchical Interactive Graphics System (PHIGS), Part 4: PHIGS Plus Lumière Und Surfaces, Draft International Standard, ISO/IEC 9592-4, 1991.
8. R. VAN LIERE (ed.). Proposal for an Improved Input Model, ISO SC 24 N353.
9. D.A. DUCE, F.R.A. HOPGOOD (eds.). New Graphical Kernel System (GKS-N) functional description, ISO/IEC JTC1 SC24 WG2 N41.

Scientific Publications

10. D.B. ARNOLD, P.R. BONO (1988). *CGM and CGI, Metafile and Interface Standards for Computer Graphics*, Springer-Verlag, Berlin Heidelberg New York.

11. CORE, Status Report of the Graphics Standard Committee, *Computer Graphics 13(3)*, August 1979.
12. T. HAGEN, P.J.W. TEN HAGEN, P. KLINT, H. NOOT (1977). *Intermediate Language for Pictures (ILP)*, preliminary report, IW 87, Mathematisch Centrum, Amsterdam.
13. M. BAKKER (ed.) (1986). *GKS Reference Manual*, 2 volumes, Centrum voor Wiskunde en Informatica, Amsterdam.
14. M. COHEN (ed.) (1986). *GKS User Guide*, Centrum voor Wiskunde en Informatica, Amsterdam.
15. G. ENDERLE, K. KANSY, G. PFAFF (1987). *Computer Graphics Programming, GKS - the Graphics Standard*, 2nd edition, Springer-Verlag, Berlin Heidelberg New York.
16. Adobe Systems, Inc., *PostScript Language Tutorial and Cook Book*, Addison-Wesley, Reading, MA, 1986.
17. R. Scheifler (ed.) (1989). *The X Protocol*, version 11.4, Massachusetts Institute of Technology.
18. P.J.W. TEN HAGEN, M.M. DE RUITER (1987). Segment grouping, an extension to the graphical kernel system. *Proceedings of Eurographics Workshop on the GKS-Review*, EUROGRAPHICS, Aire-la-Ville.
19. Proceedings of the Eurographics UK workshop at Disley on the GKS Review, The European Association for Computer Graphics, Aire-la-Ville (GE), Switzerland, 1988.
20. K.M. SINGLETON (1986). An implementation of the GKS-3D/PHIGS viewing pipeline. *Proceedings of the EUROGRAPHICS Conference at Lisbon (Portugal).*
21. D.A. DUCE, R. VAN LIERE, P.J.W. TEN HAGEN (1989). *An Approach to Hierarchical Input Devices*, Centrum voor Wiskunde en Informatica, Report CS-R8946.
22. D.A. DUCE, R. VAN LIERE, P.J.W. TEN HAGEN (1989). *Components, Frameworks and GKS Input*, Centrum voor Wiskunde en Informatica, Report CS-R8947.
23. R.H. BARTELS, J.C. BEATTY, B.A. BARSKY (1987). *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*, Morgan Kaufmann, Inc., Los Altos, CA 9402.
24. R.A. GUEDJ, H.A. TUCKER (eds.) (1979). *Methodology in Computer Graphics, Proc. IFIP WG 5.2 Workshop SEIILAC I, May 1976*, North Holland, Amsterdam.
25. S. UPSTILL (1988). *The RenderMan Companion*, Addison-Wesley.
26. G. FARIN (1988). *Curves and Surfaces for Computer Aided Geometric Design, A Practical Guide*, Academic Press, Inc.