The 11th International Conference on Geometry and Graphics, 1-5 August, 2004, Guangzhou, China

WinCAG - Education Software for Geometry

Karl-Heinz Brakhage
Aachen University of Technology, Germany

ABSTRACT This paper deals with the use of Dynamic Geometry Software for teaching. We will focus on the software package ***WinCAG***. This has been especially designed for teaching and learning (Descriptive) Geometry. It aids the visual understanding of essential aspects and interrelations for different phenomena. Modifying the viewpoint the drawings get into motion, which clarifies the shapes of the projected 3d objects. This is the key for spatial geometric cognition. Furthermore, a simple drawing can only illustrate a special situation of a circumstance, for instance Thales' Theorem. The universal validity can be recognised by changing the parameters in such a manner, that all situations appear like in a film. The changes can be done in two different ways: Dragging with the mouse or running predefined modifications in a special presentation mode. The latter one is used in Aachen as add-on for teaching Descriptive Geometry. To gain a deeper insight of the capabilities of ***WinCAG*** a demo-version can be downloaded from the author's homepage.
Keywords: CAD, Dynamic Geometry, Descriptive Geometry, Teaching

1. INTRODUCTION

For a long period geometry teaching has been conducted with a minimal amount of equipment – such as ruler and compasses. Our view is that teachers should now have at their disposal an appropriate variety of equipment from which to select. As with any approach to teaching, the educational use of information and communication technology as well as computer software needs to be well thought through and carefully planned. Computer software, particular that known as Dynamic Geometry Software, has the potential to make significant improvements in how geometry is learnt and taught.

WinCAG (**Win**dows version of **C**omputer **A**ided **G**eometry) was originally named CAG and was developed from 1985 to 1987 as a DOS-version. It has been especially designed for teaching and learning (Descriptive) Geometry. A first documentation is given in [1]. It is one of the first systems which is totally based on parametric construction. It was firstly used by students in 1987. Development stopped for more than ten years at that time. The state of the system was described in detail in [2].

In the 90s along with more powerful PCs high resolution graphics and Windows systems became available. Modifications could now be visualised *on the fly*. Our system was adapted to Windows and further features were added. The result is that both simple constructions and complicated dependencies can easily be animated. This has a strong influence on training spatial sense. The online modifications of the projection and the scaling are the keys for seeing and thinking 3d.

Let us briefly discuss the main concepts of ***WinCAG*** here. We will discuss some aspects more detailed later on. The basic elements are points, lines, circles, conics and splines. The commands for object generation can be divided into four classes: Definitions, Intersections, Tangents and operations like midpoint, parallel line, perpendicular line etc. Definitions for instance are point by its coordinates or line passing through two points. The program has no restrictions on tangents and intersections. Whenever they exist it will compute them. ***WinCAG*** internally stores the constructions with some extra information. This makes online modifications possible (using the mouse). While moving objects the new drawings for each position are consequently displayed. All geometric constructions are designed to work in all meaningful situations. The right solution is chosen by the concept of continuous changes and/or with the help of the extra information. Furthermore some special commands regarding Descriptive Geometry have been implemented. For instance there is a module to get axonometric projections from two-plane projections. The layer technique enables the teacher to visualize very complex, pre-constructed circumstances step by step not only in pictures but in motion by a few mouse clicks. Figure 1 shows an advanced stage of a complex example.

The students can clear their backlogs by examining more

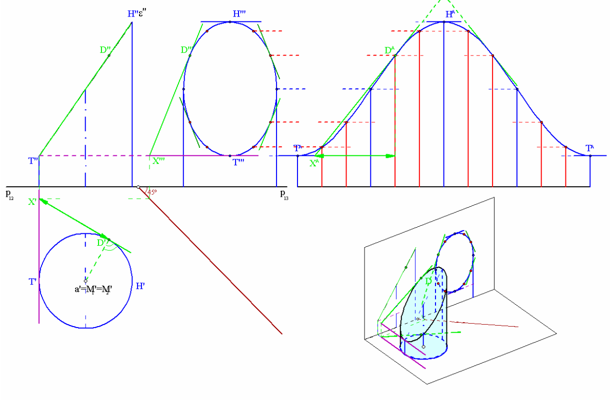

Figure 1: Section and Development of a Cylinder

examples at home in the same way. This is accomplished by demonstration-description-files. They hold the whole information for the animation – selection of layers, movement of objects etc. These are some typical examples that distinguish ***WinCAG*** from other Dynamic Geometry programs, e.g. [3]. Nevertheless these are very useful for many other situations. An "overall best system" does not exist.

WinCAG offers the possibility to export the results as bitmap or Encapsulated-Postscript-file. The experienced user can also save the construction as a text file, edit it and then reimport it to ***WinCAG***.

We will now go into more details for selected topics. Thereby we will see that ***WinCAG*** offers much more opportunities, especially for Descriptive Geometry. than other packages do on that topic.

2. COMMANDS AND MODIFICATIONS

2.1 Generating and modifying points

To clarify the concept of modification we exemplarily take a look at the points. Regarding changes we distinguish between three kinds of points

1. Points with two degrees of freedom like
   a) point by coordinates (mouse/keyboard)
   b) point relative to another point
2. Points with one degree of freedom like
   a) point on a line, a circle, a conic or a spline
   b) point given by a convex combination of two points
   c) point in a certain distance to another point and a given direction (height)
3. Fixed points like
   a) point of intersection, tangent point
   b) midpoint, nadir

Incidences can be modified afterwards. During the construction ***WinCAG*** automatically detects most of them, for instance point on a circle or intersection between two conics. Although points of the third kind can be modified, e.g. redefinition of intersections, they are not suitable for the design of demos. We need at least one degree of freedom to get the pictures into motion.

For 1.a) the command for generating a point is internally stored in a form like

P=DefinePoint(x, y),

where **x** and **y** are the (2d) coordinates of **P**. It can be dragged with the mouse or one can give new coordinates by the keyboard. But there is another more interesting way for automatic and predefined modification. The syntax of that command is

$$\mathbf{ModifyPoint(P;\ x_0,\ x_1,\ dx;\ y_0,\ y_1,\ dy).}$$

The point P will then bounce in a rectangle given by $[\mathbf{x_0,x_1}]\mathbf{x}[\mathbf{y_0,y_1}]$, stepping in x-direction with step size **dx** and in y-direction with step size **dy**. At the borders of the rectangle it bounces due to the law "angle of reflection equal to angle of incidence". Figure 2 shows an example for the above mentioned automatic modification. In Figure 3 the command can be used to automatically change the point of view (**O'**). Additionally the height of the viewer might be modified by an appropriate command according to 2.c). We will give the syntax in the next subsection.

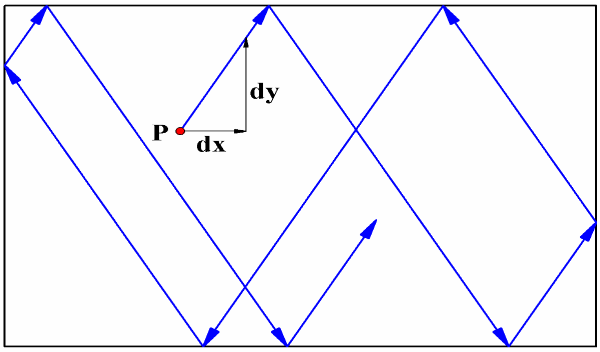

Figure 2: Automatic modifications of points

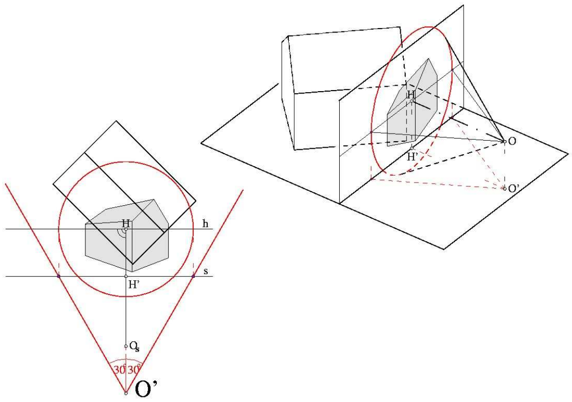

Figure 3: ModifyPoint in a practical example

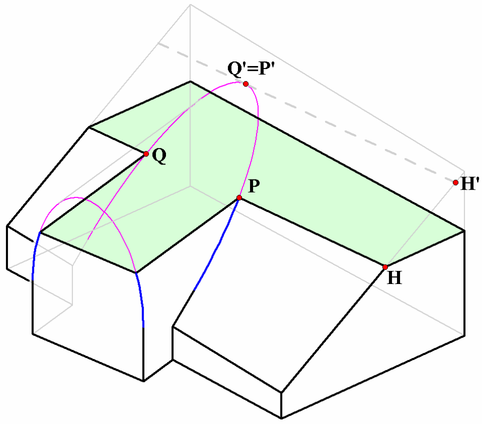

Figure 4: Resolving ambiguities

WinCAG offers a lot of possibilities to take influence on the interpretation of ambiguous conditions. The concept of continuous changes fails for instance, if the intersection of a line and a conic leads to a double root in a special situation. The scenario directly after that is undetermined. Often the user knows an extra information to resolve such situations, e.g. nearest point to wanted intersection point. Figure 4 shows a typical example for such a situation. The horizontal line at height **H'** in the roof has a double intersection with the ellipse. Thus, stepping to the height of **H** the method of continuous changes can not distinguish between **P** and **Q**. The designer can add the extra information that **P** is the point nearest to **H** and **Q** that furthest to **H** (or **P**).

2.2 The projection module

A powerful tool for spatial demonstration is the projection module of ***WinCAG***. The special arrangement of an orthographic view (top and front view) can be used to induce a coordinate system (see figure 5). This is simply done by defining the point **O'=O''**. The axes are then assumed to be vertical and horizontal respectively. The projections of a point P in top and front view are **P'** and **P''**. From these two projections we can compute the coordinates as shown in figure 5. Given an arbitrary mapping and a position **O** ***WinCAG*** is able to compute the image **P**. All coordinates of P (x, y and z) and the parameters of the projection can be used for automatic modification. With the notations of figure 5 we can state the essential commands in the following form

P'=DefinePointRelative(O', x, y),
P''=DistanceFrom(H, v; z).
DefineMap(O', O, kind, φ, θ, r, scale)
ModifyPoint(P''; z_0,z_1,dz)
ModifyMap(φ_0, φ_1, dφ; θ_0,θ_1,dθ).

The parameters **φ, θ,** and **r** represent spherical coordinates. The parameter **r** (distance) is only needed for perspective views. All other parameters should be obvious. There are further commands for the automatic modification of maps.
More complex examples are given in Figure 1 and 6. In figure 6 all parameters can be changed: The position, radius and height of the cylinder, the inclination of the cutting plane, the position of the surface line and all parameters for the view.

3 SPECIAL COMMANDS FOR DESCRIPTIVE GEOMETRY

In this section we will concentrate on some special commands for use in Descriptive Geometry. A conic section often occurs as affine or perspective image of a circle. The circumfluent square of the circle is easily mapped. ***WinCAG*** offers a special command that computes the inscribed conic section. Figure 7 shows an example. For an automatic modification P can move on an ellipse. A similar command exists for a semicircle and the corresponding rectangle
Figure 8 demonstrates the effect of a command that aids visibility checks. We call it **Horizon**, because it was originally used for perspective views. It checks a line against a collection of arbitrary bordered faces. Only the part not covered by any face is drawn. This is extremely useful for demos, because in a changing scenario these 3d effects can not be controlled by normal 2d operations. In this contents *arbitrary bordered* means that any closed curve consisting of lines, parts of circles, conics etc. can form a face.

To have maximal amount of flexibility for constructions ***WinCAG*** offers the concept of alternative commands. The effect can be seen in figure 9. The point **C** is constructed as intersection of two circles. If due to changes these circles do not intersect anymore, the alternative intersection for each circle is that with the line induced by **c**. We have additionally added a link point for **C** to keep it above **c**.

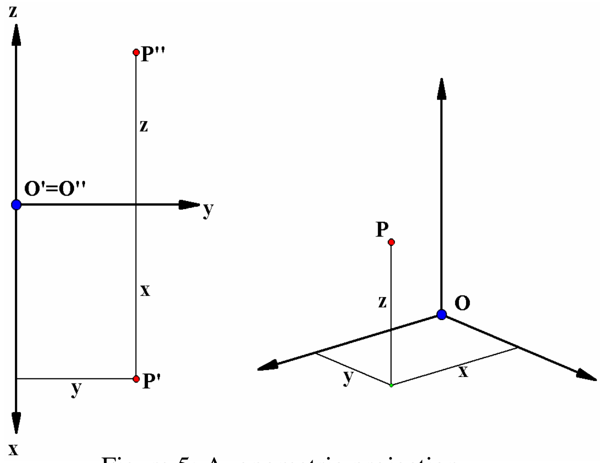

Figure 5: Axonometric projection

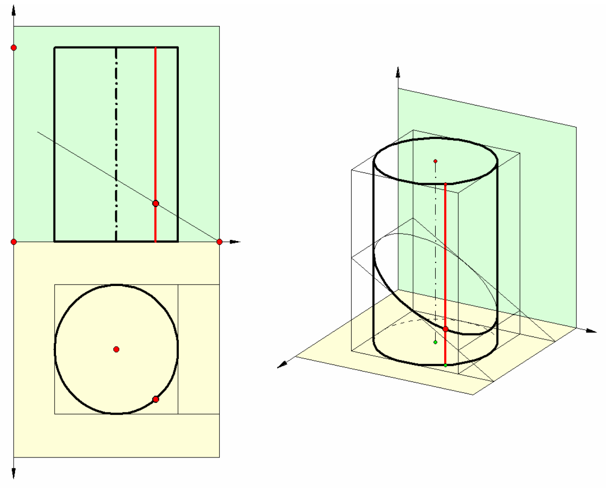
Figure 6: Using the projection module for arbitrary

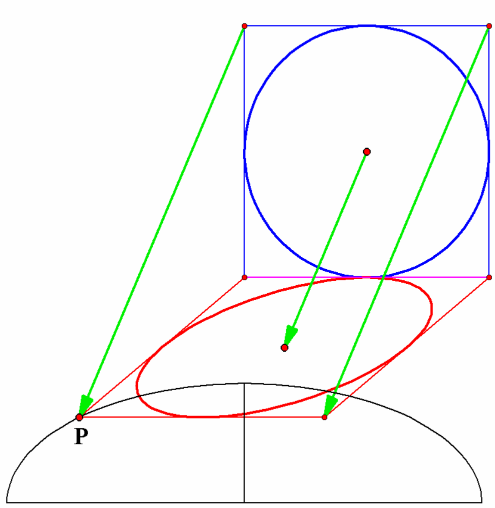

Figure 7: Special commands

We will conclude this section with yet another more complex example. We start with the orthographic projection of a building consisting of prisms, pyramids and a simple roof. We then use the projection module to get the perspective view (see figure 10). The next step is to modify the building using the circumfluent square command, tangent point-ellipse and tangent ellipse-ellipse. The intersection roof-cone is done with five points for a conic section. The perspective view is scaled a little bit to get larger.

4 CONCLUSION

In this paper only a view aspects of ***WinCAG*** and its use for improving .geometrical teaching could be shown. One has to look at real animation and online presentations to get an intrinsic overview of its capabilities. This is true for any other software on Dynamic Geometry. Such packages together with well selected demos and animations can not replace the classroom teaching, but they offer a tremendous contribution for didactical aid and to spatial geometric cognition. Due to its special design for (Descriptive) Geometry ***WinCAG*** is a mighty tool for significant improvements in teaching and learning geometry.

5 REFERENCES

1 K.-H. Brakhage, M. Nitschke, Ein zweidimensionales CAG-System zur Ingenieurausbildung in Darstellender Geometrie, Institutsbericht Nr. 47, Institut für Geometrie und Praktische Mathematik, RWTH Aachen (1987)

2 K.-H. Brakhage, Ein menugesteuertes, intelligentes System zur zwei- und dreidimensionalen Computergeometrie, Dissertation, Fortschrittsberichte VDI Reihe 20, Nr. 26(1990)

3 U. Kortenkamp, Foundations of Dynamic Geometry, Ph.D. thesis, ETH Zürich (1999)

4 K.-H. Brakhage, C. Pütz, WinCAG und der Einsatz im Fach Darstellende Geometrie für Architekten, IBDG - Informationsblätter zur Darstellenden Geometrie (2003)

ABOUT THE AUTHOR

Karl-Heinz Brakhage is member of the Institute of Geometry and Numerical Mathematics at the Aachen University of Technology. His research interests are Computer Aided Geometric Design, CAx Technologies, Grid Generation, Scientific Computing, Computer Graphics, and Development of Education Software. He can be reached by e-mail: brakhage@igpm.rwth-aachen.de, by Fax: +49(241)8092317, by phone: +49(241)8096591, the postal address: Inst. of Geometry and Numerical Mathematics / RWTH Aachen / Templergraben 55 / D-52056 Aachen, Germany, or through the Web site: igpm.rwth-aachen.de/brakhage

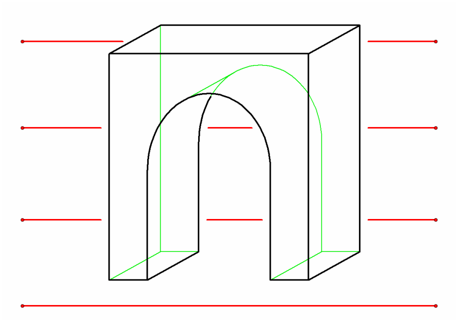

Figure 8: Visibility

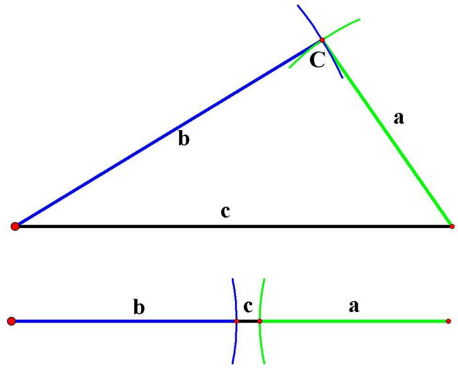

Figure 9: Alternative intersections

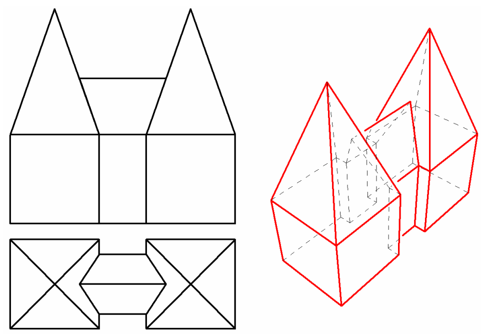

Figure 10: Orthographic and perspective view

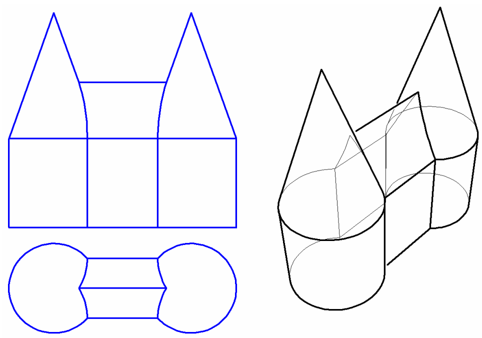

Figure 11: Modification of the building