

ONE APPLICATION OF THE INFORMATICS IN THE DESCRIPTIVE GEOMETRY

Samet Karaibryamov, Bistra Tsareva

Abstract. *This work acquaints with a program for interactive computer training to students on the subject "Mutual intersecting of pyramids in axonometry". Our software is a set of three modules, which we call "student", "teacher" and "autopilot". It gives the final solution of the problem, the traceability of various significant moments in its solution and 3D-image of the finished composition of the two intersecting polyhedra, stripped of the working lines and subjected to rotation and translation.*

Keywords: mutual intersecting, polyhedra, educational software, Descriptive Geometry, axonometry, interactive computer training.

2010 Mathematics Subject Classification: 97U70

Introduction

This scientific communication, presenting the main results of [6] acquaints with a program for interactive training on mutual intersecting of polyhedrons in axonometry. The program reproduces the manual solving of the problem where the method of auxiliary planes is used [1] [2]. Note that [6], and thus this work is the first, but a fundamental part of a project whose aim is to build software for training students on the subject "Mutual intersecting of polyhedra." This final subject in the teaching of Descriptive Geometry comprises a wide range of tasks – mutual intersection of pyramids, prisms, cones, cylinders, rotary surfaces. The solution of each problem requires a lot of time. Meanwhile, the solution includes multiple repetitions of one and the same operation which allows to optimize the learning process using a computer. The new approach has the following advantages:

- Provides a thorough study of this broad class of problems for a short time;
- Gives the solutions of the problems in all the axonometric projections;
- Allows for quick composing of new problems;
- Combines the traditional forms of teaching on Descriptive Geometry in the lectures with active self-employment in computer classes or in domestic conditions with a computer.

There are educational software tools such as Elica, GeoNext, GeoGebra, which create new ways of learning mathematics. They are primarily focused on secondary education, teacher training in it and don't issue our subject.

Preliminary

The intersection of two polyhedrons is a broken line, whose points belong simultaneously to both of the surfaces. The vertices of the intersection line are the points of intersecting of the edges of the first polyhedron with the walls of the second polyhedron and vice versa. We will name these points “pierce points” or only “punctures”. The sides of the intersection line are the intersections of the walls of the polyhedrons [1].

The approach proposed in the program is reduced to finding the intersections of each roundabout edge of one of the polyhedrons with the other one. Therefore, if one of the polyhedron has m roundabout edges and the other one has n roundabout edges, then we should repeat the task of intersecting a roundabout edge with a polyhedron ($m + n$) times. Let's denote a random roundabout edge of a polyhedron with \mathbf{g} . Then the solution of the problem of intersecting \mathbf{g} with the other polyhedron passes through the following steps: A) Defining a plane α incident with the line \mathbf{g} ; B) Finding the intersection of the plane α with the polyhedron; B) Determining the points of intersection of \mathbf{g} with the intersection found in the previous step. The plane α incident with the line \mathbf{g} is called auxiliary plane.

Obviously, in step A) there are countless possibilities in choosing the plane α . Taking this opportunity, we simplify as much as possible step B) in the solution as well as the construction of the traces of α . When the polyhedron is a pyramid, we choose an auxiliary plane incident with the apex V of the pyramid. When the polyhedron is a prism, we choose an auxiliary plane parallel to the roundabout edges of the prism. It's known that in the first case the intersection is a triangle and in the second case it's a parallelogram. The points of Intersection of \mathbf{g} with the triangle or whit the parallelogram are the pierce points of \mathbf{g} with the polyhedron.

As we know [1]:

a) The lines $\mathbf{m}_\alpha = \alpha \cap \mu$, $\mathbf{n}_\alpha = \alpha \cap \nu$, $\mathbf{p}_\alpha = \alpha \cap \pi$ are called respectively the first, the second and the third traces of α ;

b) The points $\mathbf{M}_t = \mathbf{t} \cap \mu$, $\mathbf{N}_t = \mathbf{t} \cap \nu$, $\mathbf{P}_t = \mathbf{t} \cap \pi$ are called respectively the first, the second and the third steps of \mathbf{t} ;

c) $\mathbf{t} \in \alpha$ if and only if both of the conditions $\mathbf{M}_t \in \mathbf{m}_\alpha$, $\mathbf{N}_t \in \mathbf{n}_\alpha$, $\mathbf{P}_t \in \mathbf{p}_\alpha$ are true.

In the problems we discuss the bases of the polyhedra lie in the coordinate planes μ , ν , π of the spatial coordinate system $Oxyz$, in respect of which the objects are set [3].

Hence: the steps of the roundabout edges of the polyhedron coincide with their basic vertices lying in the coordinate planes; the plane of the base of the polyhedron intersects the auxiliariy planes through the roundabout edges in their corresponding traces.

§1. Mutual intersecting of two convex polyhedrons

This section gives information about those components of the program which are common for all three cases of two polyhedra - two pyramids, pyramid and prism, two prisms.

The program is written in C # [4] [5]. The program provides a choice of one of the five popular axonometric projections – cabinet perspective, cavalier perspective, military perspective, increased orthogonal dimetry, increased orthogonal isometry. This concerns the construction of the axonometric cross and the ability to recalculate the coordinates of a point as follows: $A(x, y, z) \rightarrow A'(px, qy, rz)$, where p, q, r are the coefficients of variation.

The next step in the solution is describing the edges. The program provides the well-known rules for determining the visibility of the edges of the polyhedron [1].

Central place in the program is assigned to finding the pierce points of each roundabout edge of one polyhedron with the other polyhedron. The program describes the following four cases :

P1) The roundabout edge of one of the polyhedra intersects the other one in two points located in two different walls of the other polyhedron;

P2) The roundabout edge of one of the polyhedra intersects the other one in two points, one of which lies in the interior of one of the walls and the other point is on a roundabout edge;

P3) The roundabout edge of one of the polyhedra touches the other one, i.e. they have one common point which is the point of intersection of the considered roundabout edge with some roundabout edge of the other polyhedron

P4) The roundabout edge of one of the polyhedra has no common points with the other polyhedron.

The next milestone in the solution of the task is connecting the pierce points we've already found, to obtain the intersection line. We call this stage a construction of the pierce scheme. For its realization the following three functions are introduced: **UnwindProbodTable** (determines the sequence of the pierce points in the intersection line); **CreateLines** (creates the segments of the intersection line, and sets their visibilities); **CreateProbodSchemeImage** (creates a graphical representation of the pierce scheme). These functions use the program's internal pierce table. The table is filled by calling the appropriate functions of the following: **Hedra1IntersHedra2** (adds a pierce point of a roundabout edge of the first polyhedron with the roundabout wall of the second one); **Hedra2IntersHedra1** (adds a pierce point of a roundabout edge of the second polyhedron with a roundabout wall of the first one); **CrossPoint** (adds a pierce point of a roundabout edge of the first polyhedron with a roundabout edge of the second one) for each pierce point.

At the end the program adjusts the visibility of the edges of both of the polyhedra in 4 stages taking into account their location relative to one another and the fact that the segment between the pierce points of a roundabout edge is invisible.

An interesting component of the program is the three-dimensional representation of the finished composition of the two intersecting polyhedrons without the working lines and the ability to change the view by rotation and translation of the composition.

Note that the problems which illustrate the operation of the program are presented with: a) the final decision, including implementation of all constructs b) intersection of two polyhedrons after hiding the working lines c) pierce scheme.

§2. Mutual intersecting of two pyramids

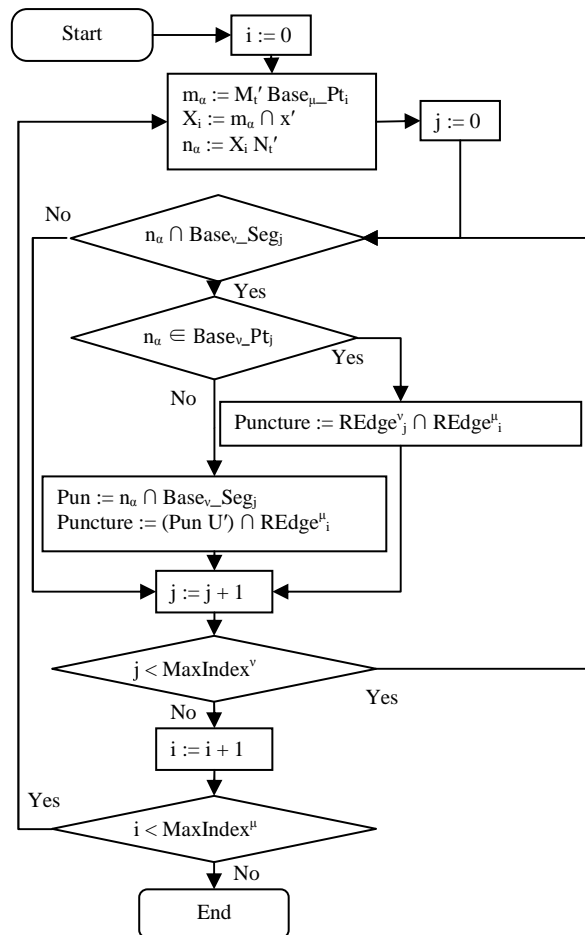
Let's accept the following terms: the vertices of the first pyramid we denote by the letters V, A, B, C, D,... where A, B, C, D,... are the basic vertices; the vertices of the second pyramid will be denoted by the letters U, P, Q, R, S,... where P, Q, R, S,... are the basic vertices.

Construct the peak line $t = UV$. All auxiliary planes pass through t , which ensures that their intersections with the pyramids are triangles. Locate the steps of the apex line which are necessary for subsequent constructs.

$$M_t' = t' \cap t_1, N_t' = t' \cap t_2, P_t' = t' \cap t_3, \text{ where } t' = V'U'$$

$$t_1 = V_1U_1, t_2 = V_2U_2, t_3 = V_3U_3$$

A flowchart describing in short how to find the pierce points of the roundabout edges of a pyramid which base is on μ with a pyramid which base is on ν .

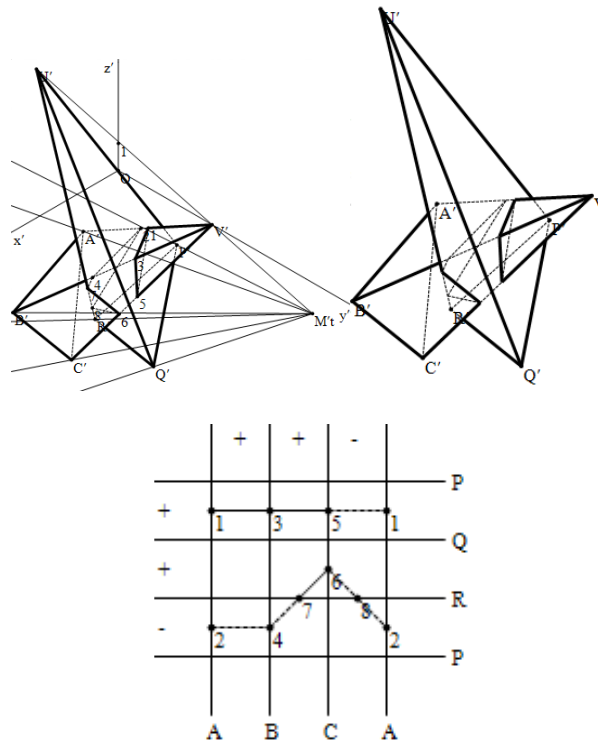


In the above flowchart “REdge” stands for “roundabout edge”, “MaxIndex” is the number of sides of the base of the respective polyhedron.

We present the operation of the program with the following three examples where we’ve used only the main axonometric images for the vertices of the bases and the intersection line.

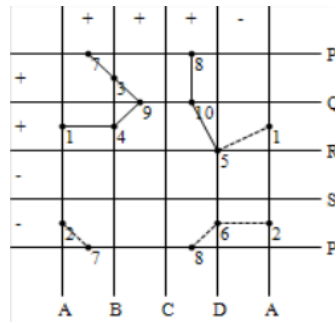
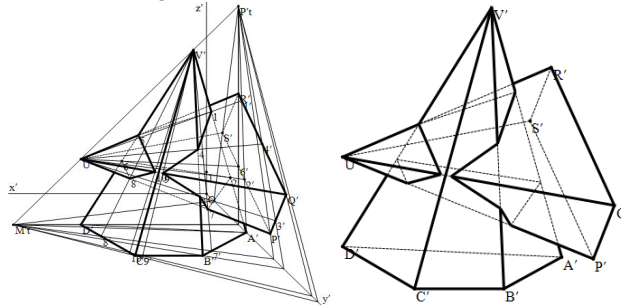
2.1 *Both bases of the pyramids are in the coordinate plane μ .* Now it’s sufficient to have only the first step of $t - M_t$, on the figure. Auxiliary planes are joined in the sketch only with their first traces. Their main axonometric images pass through the point M_t' and the basic vertices of the two pyramids:

Example. 1. [3] In orthogonal isometry represent the mutual intersecting of the pyramids $VABC$ and $UPQR$ [$V(2; 6; 2), A(3; 1,5; 0), B(7,5; 3; 0), C(8; 6; 0)$], [$U(3,5; 0; 5,5), P(1,5; 4; 0), Q(6,5; 8; 0), R(6; 5; 0)$].



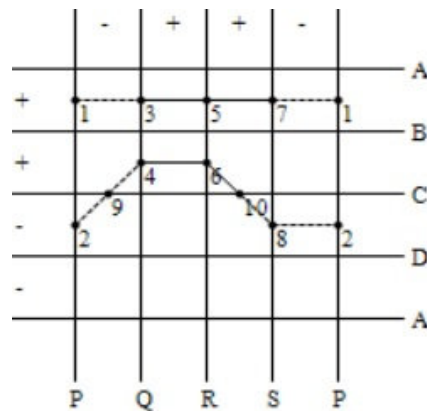
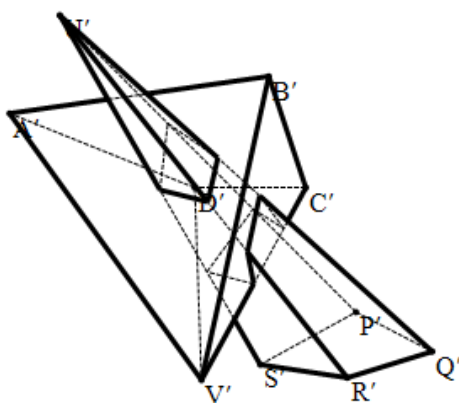
2.2 *One pyramid has a base in μ , and the other one has a base in π .* Now the first and the third steps of the apex line are needed, because the auxiliary planes will be involved with their first and third traces. Here we have selected a problem in which all four of the described cases of pierce points P1), P2), P3), P4) appear.

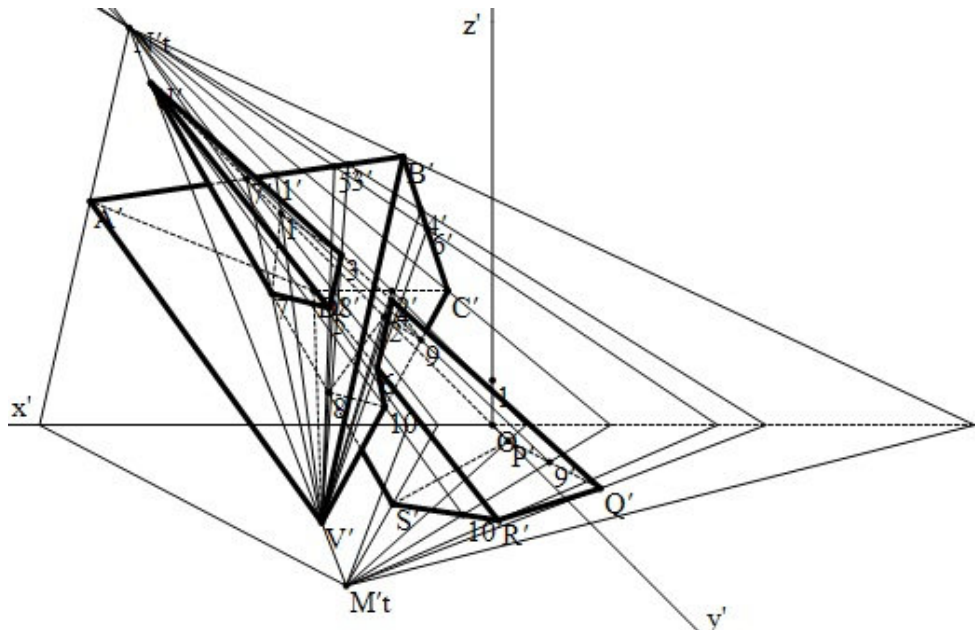
Example. 2 [3] In Cavalier perspective represent the mutual intersecting of the pyramids $VABCD$ и $UPQRS$ [$V(2; 2; 8)$, $A(0; 2,5; 0)$, $B(3; 4; 0)$, $C(6; 4; 0)$, $D(7; 2; 0)$], [$U(7; 2; 3)$, $P(0; 4; 1)$, $Q(0; 5; 3,5)$, $R(0; 2; 6)$, $S(0; 1; 3,5)$].



2.3 One pyramid has a base in μ , and the other pyramid has a base in ν . Now the first and the second steps of the apex line are needed because the auxiliary planes will be included with their first and second traces.

Example. 3. In cabinet perspective represent the mutual intersecting of the pyramids $VABCD$ и $UPQRS$ [$V(7; 9; 1)$, $A(9; 0; 5)$, $B(2; 0; 6)$, $C(1; 0; 3)$, $D(4; 0; 3)$], [$U(8; 1; 8)$, $P(0; 1; 0)$, $Q(-1; 4; 0)$, $R(2; 6; 0)$, $S(4; 5; 0)$].





The first and the third examples are created with the help of our program.

References

- [1] G Petrov, Descriptive Geometry, 1960, DI, Tehnika, Sofia, 1960.
- [2] D. Chorbadjiev, Descriptive Geometry, Sofia, 2002.
- [3] S. Georgiev, S. Dimitrov, Problems book on Descriptive Geometry, DI, Tehnika, Sofia, 1973.
- [4] J. Sharp, Microsoft Visual C# 2008 Step by Step, 2008, Microsoft Press.
- [5] R Stephens, C# Graphics Programming, 2008, Wiley Publishing.
- [6] S. Karaibryamov, One application of informatics in Descriptive Geometry, Thesis, FMI of the University of Plovdiv, 2010.

Samet Karaibryamov, Bistra Tsareva
 Faculty of Mathematics and Informatics
 University of Plovdiv
 236 Bulgaria Blvd.
 4003 Plovdiv, Bulgaria
 e-mail: einismotic@hotmail.com, btsareva@uni-plovdiv.bg

