Proceedings of Algoritmy
2000, pp. 1–10

VISUALIZATION OF LEVEL SETS AND SINGULAR SURFACES IN DIFFERENTIAL GAMES *

V.L. AVERBUKH†, S.S. KUMKOV†, O.A. PYKHTEEV†, AND D.A. YURTAEV†

Abstract. In the paper, algorithms and programs of scientific visualization carried out in the Institute of Mathematics and Mechanics, Russian Academy of Sciences (Ekaterinburg, Russia) are considered. The objective field of these programs is the linear differential games with fixed terminal time, convex terminal payoff and geometrical constraints on the player controls. The main element of the solution of a game (and the object for visualization) is the value function of the game defined by a collection of its level sets. Also, it is useful to investigate (including visual approach) the singular surfaces of the game, that is the surfaces where the value function has some peculiarities.

Screenshots of the programs and pictures of views produced by them are shown.

Key words. scientific visualization, differential games, value function, singular surfaces, OpenGL, ARCBALL-controller

AMS subject classifications. 68U05, 90D25, 49N55

1. Introduction. Dynamics of a differential game is usually described [1, 2, 3] as $\dot{x} = F(t, x, u, v)$. Here t is the time, x is the phase vector (possibly, multidimensional), u and v are the controls of the first and second players. These controls are taken from compacta P and Q, correspondingly. The case is quite typical when the terminal time T is fixed and the payoff function $\varphi(x)$ is determined at the terminal instant. The first player manages his control to minimize the payoff φ, the second one maximizes it. The players controls are of the feedback type. For the accurate definitions, one can see [1, 2, 3].

For a quite wide class of games, the best guaranteed results of players are equal. Corresponding common function $(t, x) \mapsto V(t, x)$ is called the value function of the game. The construction of the value function is the essential part of solving a differential game.

There are a number of ways for presentation of the value function. One of them has the form of a collection of level sets of the value function. A level set of the value function corresponding to the parameter c is the set where the function is not greater than c. For the games with the fixed terminal time, each level set of the value function has a certain stability feature and is a maximl u-stable bridge [2, 3]. For the games of this type, the level sets of the value function can be imagined as “tubes” (finite or infinite in time) along the time axis. In this paper, the terms “level set”, “stable bridge” and “tube” denote the same object.

The games of a very narrow class can be solved analytically. Thus, numerical methods are widely used [4, 5, 6].

A partial case of differential game is the game with linear dynamics:

$$\dot{x} = A(t)x + B(t)u + C(t)v, \qquad \begin{array}{l} t \in [t_0, T],\ x \in R^n, \\ u \in P,\ v \in Q,\ \varphi(x(T)). \end{array} \tag{1.1}$$

†Institute of Mathematics and Mechanics, Ural Branch of RAS, S.Kovalevskaya str. 16, Ekaterinburg, 620219, Russia, e-mail: averbukh@imm.uran.ru

*This research was supported by the Russian Foundation for Basic Researches under Grants No. 00–01–00358, No. 99–07–90441 and No. 99–07–90078.

Here, $A(t)$, $B(t)$ and $C(t)$ are matrices of appropriate dimensions. If additionally, the payoff function φ is convex, then the value function $V(t,x)$ is convex on x at any instant t. In the Institute of Mathematics and Mechanics, Ural Branch of the Russian Academy of Sciences (Ekaterinburg, Russia), since the beginning of 80's, numerical algorithms and programs for solving linear differential games with fixed terminal time and geometrical constraints on player controls are developed [6, 7].

It is known [3] that when the payoff function depends only on two components x_i, x_j of the phase vector at the terminal instant, the substitution $y(t) = X_{i,j}(T,t)x(t)$ provides the transformation to the equivalent differential game of the second order:

$$\begin{gathered} \dot{y} = D(t)u + E(t)v, \\ t \in [t_0, T], \quad y \in R^2, \quad \varphi(y(T)), \quad u \in P, \quad v \in Q, \\ D(t) = X_{i,j}(T,t)B(t), \quad E(t) = X_{i,j}(T,t)C(t) \end{gathered}$$

Here $X_{i,j}(T,t)$ is a matrix combined of two corresponding rows of the fundamental Cauchy matrix. The time interval, the players control constraints P and Q and the payoff function φ stay the same as in the original linear game (1.1).

After that, each tube (or a maximal stable bridge, in other words) is a three-dimensional body in the space of two new phase coordinates y_1, y_2 (the equivalent coordinates) and the time t. An effective investigation of the problem is possible by means of visual exploration of the tubes constructed numerically.

Another object interesting for a researcher within the framework of differential games is the singular surfaces, that is the surfaces where the value function and optimal motions have some peculiarities. In the theory of differential games, there is a classification [1] of such surfaces. The classification is based on the behavior of the optimal motions on the surfaces and near them. In Fig.1.1, some types of the surfaces are represented. As for the dispersal surface, two optimal motions go out from any its point. One motion goes to one side from the surface, the second one goes to the other

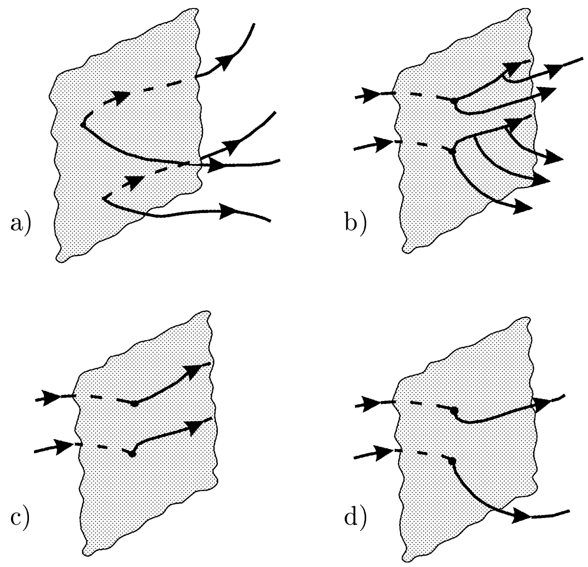

FIGURE 1.1. *Behaviour of the optimal motions near singular surfaces of different type: a) dispersal surface; b) equivocal surface; c) switching surface without leaving; d) switching surface with leaving.*

side. When a motion reaches an equivocal surface, it splits into two motions. One of these motions leaves the surface, the second one goes upon it. The latter motion also splits in the same way at each future instant. Switching surface gives fracture to optimal motions, but without splitting. There are two types of switching surfaces: with leaving and without it.

Many difficulties appearing when solving differential games are due to the presence of singular surfaces. For the games of the kind (1), singular surfaces are also put into the three-dimensional space. Therefore, they also can be effectively visualized. Algorithms for numerical detection and classification of the singular surfaces for the games, where the player controls are scalar, are developed [9, 10] in the middle of 90's. Nowadays, such algorithms for the case of general convex control constraints are under construction.

The paper presents programs for visualization of the value function level sets and the singular surfaces in the linear differential games.

2. Visualization of stable bridges. Computational programs [6, 8] for constructing a stable bridge are based on backward procedures similar to ones carried out in dynamic programming. The result of these programs is a collection of t-sections of a stable bridge. Each of these sections is a convex polygon and is defined by coordinates of its vertices (two equivalent coordinates y_1, y_2 and the time coordinate t).

Visualization approach used in the 80's was quite trivial. Using it, the screen picture contains projections to the equivalent coordinates of a number of different t-sections of a level set. Figure 2.1 presents an example of this type image. The another type is to draw projections t-secions (corresponding to the same time instant) of a number of level sets. Both visualizations are not satisfactory, because with increasing number of sections drawn in the screen, the informativeness of the picture is lost.

So, some new visualization programs have to be developed which has to provide a convenient tool for investigator.

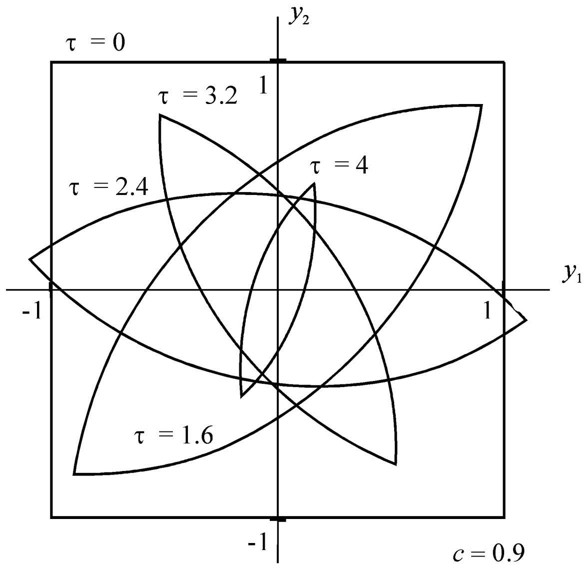

FIGURE 2.1. *Example of old-style visualization: for a given value of c a number of sections corresponding to different backward time t instants are shown.*

Rejection of usage of some standard environments of scientific graphics and the idea of creation of specialized program were caused by a score of reasons, among which are

- the presence of special data format files with the information produced by the computational program;
- the necessity of the special procedure for constructing a level set surface, which is able to underline interesting peculiarities of the surface;
- the especial demands to the interactivity of the user interface, particularly, convenient and quick changing parameters of the view.

Such a program was created by authors [11, 12]. The program developed gives the image of each tube in general as it is a three-dimensional body. Surface of the body is constructed on the basis of separate t-sections produced by computational program.

The test version was created for the UNIX operational system with the X-Window graphic shell. Further, the program was rewritten for MS Windows 9x/NT 4.0 in Delphi. As the basic graphic library, the OpenGL was used.

One of the main modules of the program is the procedure of constructing the surface of the tube on the basis of separate t-sections. The algorithm connects each two neighbor sections by triangles. Firstly, "sharp" vertices are detected. A vertex is sharp if this vertex angle is less than some threshold (which is chosen empirically). Then, the corresponding sharp points of neighbor t-sections are connected. The correspondence is established by the geometrical proximity. With that, the boundaries of these polygons are divided apart to a number of arcs by these sharp vertices. After that, pairs of corresponding arcs of neighbor t-sections are triangulated, that is the space between them is tiled by triangles (see Fig. 2.2). This triangulation algorithm allows to underline nonsmoothnesses of the tube surface. These nonsmoothnesses are very interesting because they are connected to singularitites of the value function.

The screenshot of the main window of the program is shown in Fig. 2.3. The dialog with the basic control elements is the separate window. After the start of the program, the user loads one or more files. Each of them contains information on single level set (in a certain format). During the work session, more files can be loaded. When a file is read, the program processes the information from it and builds the surface of the tube.

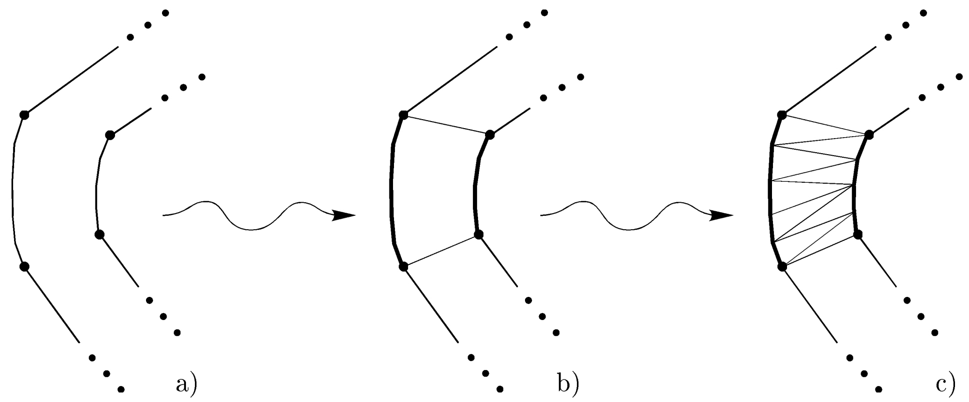

FIGURE 2.2. *Illustration to the triangulation process: a) search for "sharp vertices" (found sharp vertices are shown by small circles); b) connecting corresponding sharp vertices; c) triangulating between corresponding arcs.*

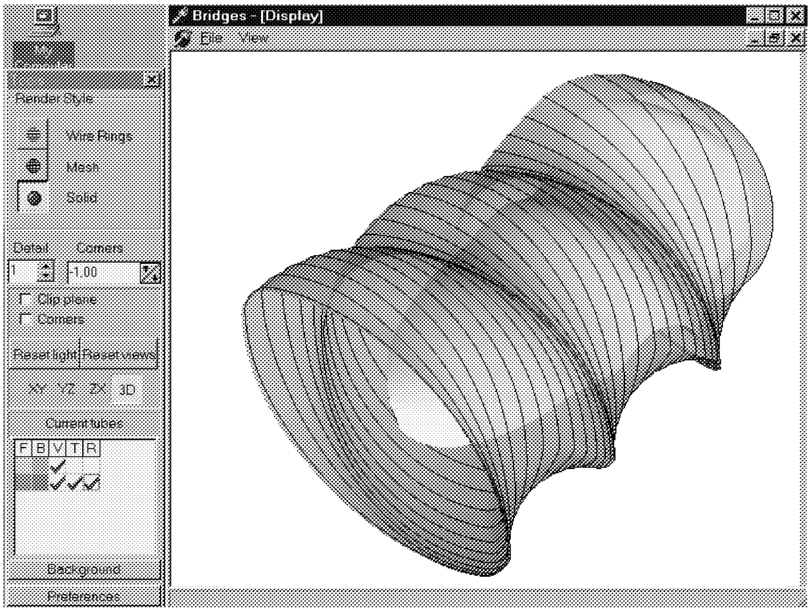

FIGURE 2.3. *The screenshot of the main window of the program for visualization of level sets.*

The program has the following features:

1. Switching the drawing regimes:
   - the wirering regime (Fig. 2.4a): only t-section contours are drawn (possibly, with decimation);
   - the mesh regime (Fig. 2.4b): the triangle mesh is drawn;
   - the solid regime (Fig. 2.4c): the tube is drawn as a solid body lighted by a dot radiant. The Gouraud shading is used.

In solid regime, picture refresh takes a time (from 2 upto 10 seconds depending on the concrete scene). So, the wirering regime is useful for seeking good point of view.

2. Rotation and moving the scene.

Rotation of the scene is made by means of the *ARCBALL-controller* algorithm [13]. Its idea is in the following. By click of a mouse button, the user fixes the projection of a point from an imaginary three-dimensional sphere. Further, the projection follows the mouse and, consequently, the sphere point also moves. Due to this motion, the sphere rotates and the scene rotates correspondingly to the sphere rotation.

The scene can be moved orthogonally to the line of sight.

3. Representation of the scene as a three-dimensional picture and as a projection into coordinate planes.
4. Movement of the light source. The movement is implemented also by the *ARCBALL* algorithm.
5. For each tube independently, the following attributes can be set: visibility/invisibility, opaqueness/transparency, the presence of section contours.

Changing two last attributes is useful when two or more embedded tubes are investigated at the same time (see Fig. 2.5).

6. Cutting a part of the scene by a plane parallel to the time axis. The plane can be moved by the user, but cannot be rotated.

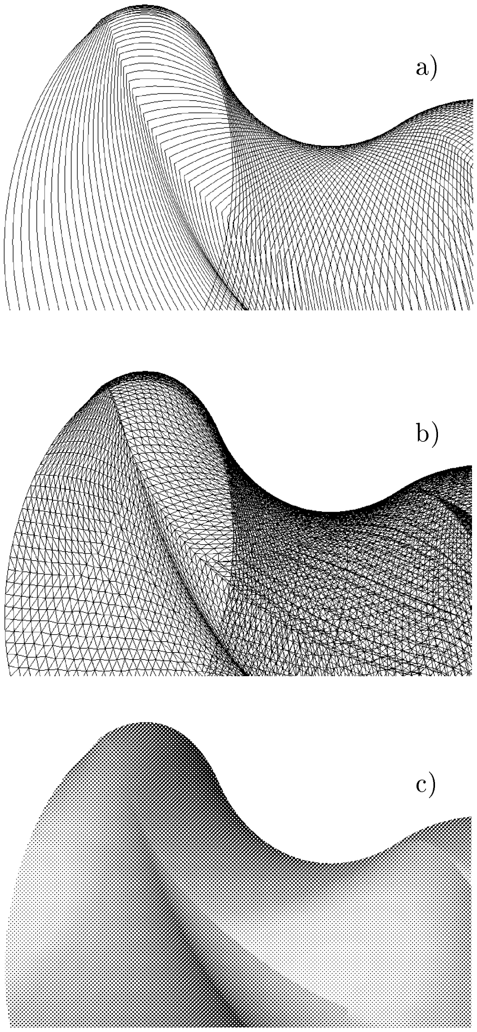

FIGURE 2.4. *Possible drawing: a) the wirering regime, b) the mesh regime, c) the solid regime.*

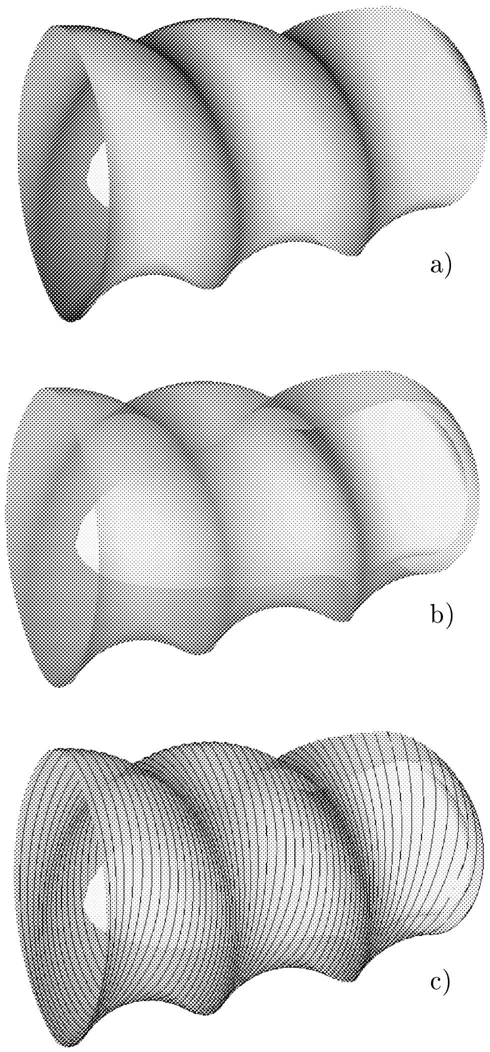

FIGURE 2.5. *Possible regimes of two embedded surfaces view: a) both surfaces are opaque, b) the outer surface is transparent, c) the outer surface is transparent and some its contours are shown.*

Application of this feature is useful when three or more tubes are shown simultaneously or the tubes have a complicated shape (see Fig. 2.6). In this case, the transparency feature sometimes does not give a recognizable view.

Figures 2.7 and 2.8 demonstrate the peculiarities of the level sets, which are interesting for researcher. Figure 2.7 contains a picture of a line of nonsmoothness on the surface of a level set. Presence of such a fracture is the sign of nonsmoothness of the value function itself and, consequently, of a peculiarity in the solution of the game. A fragment of a level set with a narrow "throat" is shown in Fig. 2.8. There, the level set degenerates almost to a point. Adequate representation of the structure of the level sets near such a "throat" needs very precise computations.

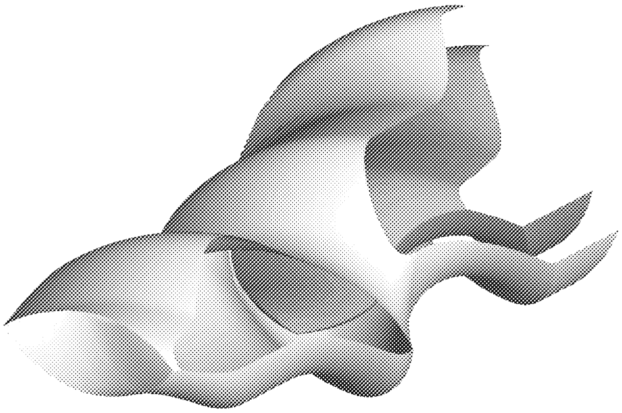

FIGURE 2.6. *Two complicated tubes cut by a plane.*

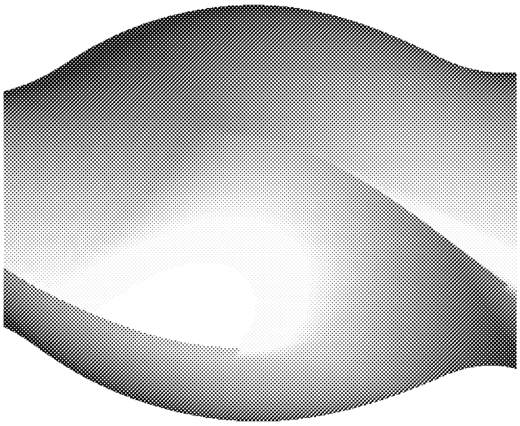

FIGURE 2.7. *Nonsmoothness of the level set surface.*

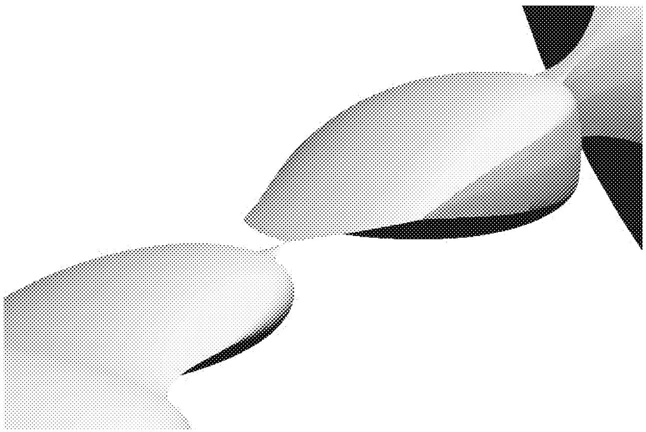

FIGURE 2.8. *A narrow "throat" of a level set.*

So, a specialized program for visualization of the value function level sets is developed. This program is oriented to specific data format produced by existing computational programs. The software created is quite convenient, contains only necessary tools and its executable file is very compact. The researcher can investigate level sets of the value function as physic bodies, that is explore in detail their geometrical configuration, mutual placement, presence of nonsmothnesses on their surface, presence of narrow "throats".

3. Visualization of singular surfaces. The construction of singular surfaces is based on the detection and classification of singular points on the t-sections of stable bridges. The block of detection and classification of the singular points is built into the procedure of backward construction of new t-section of the stable bridge. At the end of the process of construction of the stable bridge, the information on separate singular points is included into the corresponding output file. For each point, its coordinates (two geometric coordinates and the time instant) and the type of singularity are specified. For the surface construction, the points computed for a collection of the bridges are necessary. Besides, the more dense the grids of values of the value function and the time are taken, the more precisely it would be possible to reconstruct the surfaces.

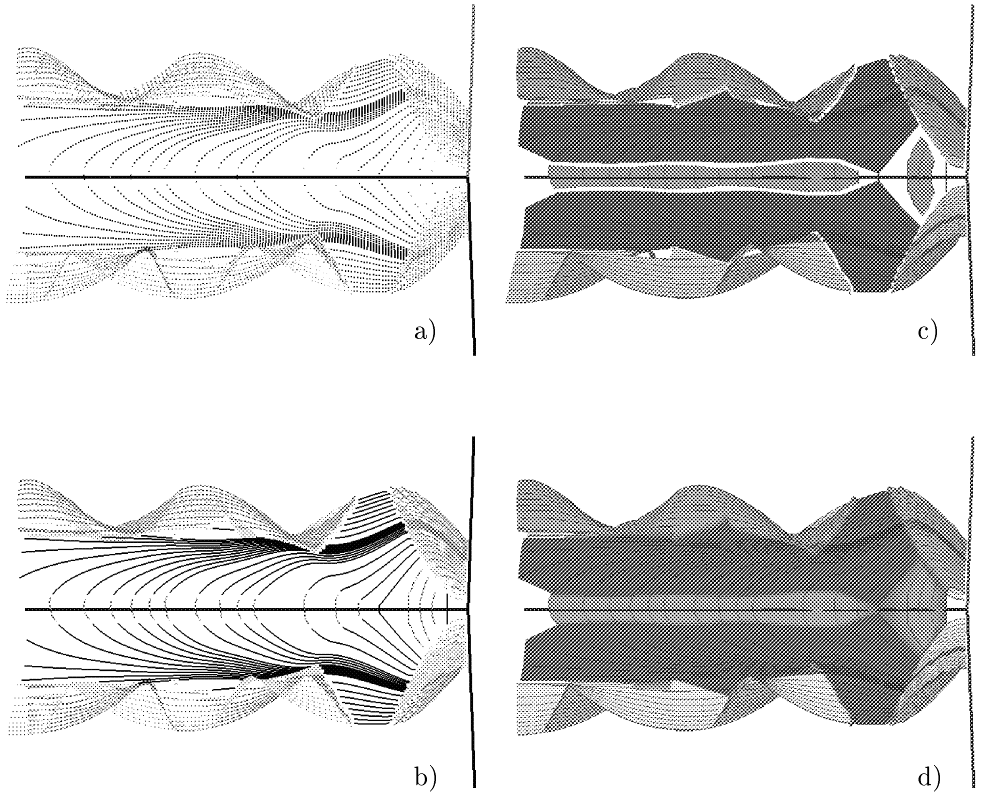

FIGURE 3.1. *Illustration of the process of reconstruction of the singular surfaces: a) separate points taken from the collection of bridges; b) the points of the same type that lie on the same bridges are joined into the lines; c) the lines of the same type that lie on the neighbor bridges are joined into pieces of the surfaces; d) separate pieces are united into the result surface. Here, the horizontal line is the time axis, two other straight lines are the axes of the equivalent coordinates.*

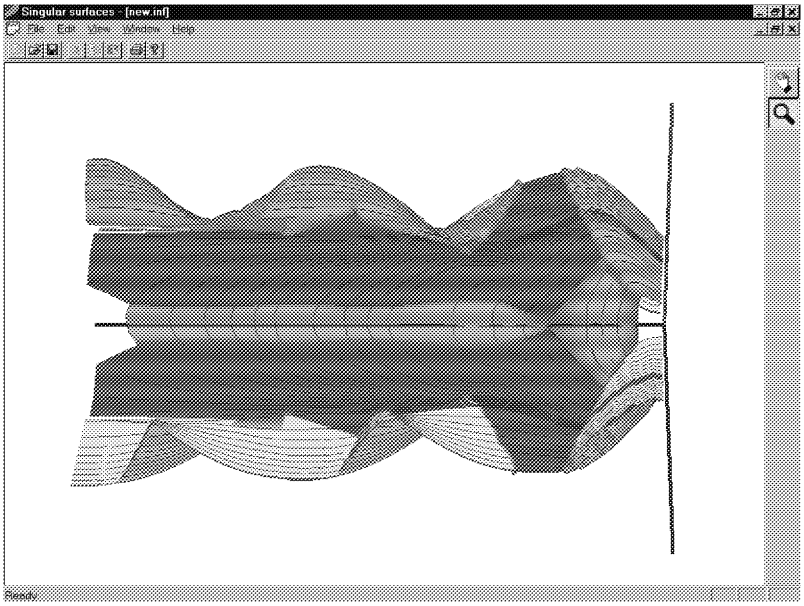

FIGURE 3.2. *Screenshot of the main program for visualization of the singular surfaces.*

In the program for visualization of the singular surfaces, the crucial fragment is the procedure of reconstruction of the surface on the basis of a collection of its points. Now the following method of reconstruction is realized. At the beginning, the points (Fig. 3.1a), that have the same type of singularity and are taken from the neighbor t-sections of a bridge, are joined into lines (Fig. 3.1b). Further, the lines of the same type, that lie on the neighbor bridges, are joined into "pieces" of the surfaces (Fig. 3.1c). At last, these pieces are united into the result surface (Fig. 3.1d). It can be seen that such procedure is absolutely empiric, since, from the algorithm of calculation of each separate point, it is impossible to extract any information on the relation between the points that lie on the same tube. As a result, the constructed images, giving the general view of the singular surfaces, sometimes can have some faults at the places of concatenation of separate pieces.

The screenshot of the main window of the program is shown in Fig. 3.2. The system discussed is a probational one with restricted capabilities in comparison with the program described in the previous section. The user is provided by the following tools: to rotate the object, to zoom in/out the scene, to switch on/off various elements of the scene (points, lines, separate parts, junctions between pieces, etc.)

Singular surfaces can be treated as the "skeleton" of the solution of a differential game. The specialized visualization program developed allows to study effectively the structure of singular surfaces, to investigate the dependence them on the parameters of a problem (including the case of changing the singularity type).

4. Results of the experimental application. Conclusions. The programs discussed are used now for more than a year period. The following properties of the programs are investigated in details: convenience of the user interface, clearness of visual representation, possibility of application to other research problems. The following conclusions on positive and negative features of the programs have been made.

Conclusion 1. Application of specialized software systems for visualization of research results seems to be very fruitful. Integral comprehension and interpretation of large presented data are significantly facilitated. These visual tools are useful for search and investigation of peculiarities of differential games. The usage of the lighting distinction of the three-dimensional structure is very important.

Conclusion 2. Specialized systems for visualization have a score of advantages in comparison with standard graphic and scientific packages: strictly proposed algorithms for constructing surfaces to be visualized, data formats matched with computational programs, presence of all necessary tools for image manipulating and only them, relatively small executable file.

Conclusion 3. The collections of tools realized in the discussed programs (orthogonal projecting, one source of light, fixed orientation of a single cutting plane) often are non-sufficient for satisfactory investigation of the structure of the objects to be visualized.

Conclusion 4. The collections of tools realized in the discussed programs are generally the same. The difference is in the procedures of constructing the objects to be visualized.

Conclusion 5. Modules of the reconstruction must have an opportunity to use different procedures of reading the input data. It is stipulated by the fact that some evolutions of the computational programs take place, and as a result, formats of their output files also change.

The accumulated experience and topics of the conclusions are taken into account in elaboration of a new version of the visualization program. This program contains only a collection of instruments for an image drawing and manipulation. New tools for manipulation will be included.

Surfaces to be visualized by this program will be constructed by independent modules that are registered as dynamically linked libraries. In their turn, the procedures for surface reconstruction will similarly use independent modules of the data reading. Unified interface of such modules will permit to change the libraries if the data format or the reconstruction algorithms are changed. The variation of these modules will not cause recompilation of the visualization module.

REFERENCES

[1] R. Isaacs, *Differential games*, John Wiley and Sons, New York, 1965.

[2] N.N. Krasovskii and A.I. Subbotin, *Positional differential games*, Nauka, Moscow, 1974 (in Russian).

[3] N.N. Krasovskii and A.I. Subbotin, *Game-theoretical control problems*, Springer-Verlag, New York, 1988.

[4] V.N. Ushakov, *Construction of solutions in differential games of pursuit-evasion*, in Differential Inclusions and Optimal Control, Lecture Notes in Nonlinear Analysis, Vol. 2, Nicholas Copernicus University, Torun, 1998, pp. 269–281.

[5] M. Bardi, T.E.S. Raghavan and T. Parthasarathy, (eds.), *Stochastic and Differential Games: Theory and Numerical Methods*, Vol. 4 of Annals of the International Society of Dynamic Games, Birkhäuser, Boston, 1998.

[6] A.I. Subbotin and V.S. Patsko, (eds.), *Algorithms and programs for solving linear differential games*, Institute of Mathematics and Mechanics, Sverdlovsk, 1984 (in Russian).

[7] A.I. Subbotin, *Constructive theory of positional differential games and generalized solutions to Hamilton-Jacobi equations*, in ref. [5], pp. 3–67.

[8] S.S. Kumkov and V.S. Patsko, *Level sets of value function and singular surfaces in linear differential games*, in A Proceedings Volume from the IFAC Workshop on Nonsmooth and Discontinuos Problems of Control and Optimization, Chelyabinsk, Russia, June 17–20 1998,

V.D.Batukhtin, F.M.Kirillova and V.I.Ukhobotov, eds., Pergamon Press, Great Britain, 1999, pp. 143–148.

[9] S.S.Kumkov and V.S.Patsko, *Parallel algorithm for solving linear differential games. Analysis of singular surfaces*, in Proceedings of the Eighth International Colloquium on Differential Equations, August 18–23, Plovdiv, Bulgaria, 1997, D. Bainov, ed., Utrecht, the Netherlands, 1998, pp. 275–284.

[10] S.S.Kumkov and V.S.Patsko, *Construction of singular surfaces in linear differential games*, in Proceeding Volume of 8th International Symposium on Dynamic Games and Applications, July 5–8, 1998, Chateau Vaalsbroek, Maastricht, the Netherlands, pp. 341–347.

[11] V.L.Averbukh and D.A.Yurtaev, *An approach to development of specialized visualization systems for the problem of stable bridges constructing in linear differential systems*, in Algorithms and Program Tools for Parallel Computations, Vol.2, L.D.Popov and S.V.Sharf, eds., Inst. of Math. and Mech. UrB RAS, Ekaterinburg, Russia, 1998, pp. 3–9 (in Russian).

[12] V.L.Averbukh, S.S.Kumkov, V.S.Patsko, O.A.Pykhteev and D.A.Yurtaev, *Specialized Visualization Systems for Differential Games*, in Progress in Simulation, Modeling, Analysis and Synthesis of Modern Electrical and Electronic Devices and Systems, N.E.Mastorakis, ed., WSES Press, 1999, pp. 301–306.

[13] K.Shoemake, *ARCBALL: A user interface for specifying three-dimensional orientation using a mouse*, in Proceedings of Graphics Interface'92, Canadian Information Processing Society, Toronto, 1992, pp. 151–156.