Design of Parameterized Model for a 3D Dynamic Geometry System

[1]Zheng Liu, [2]Guanhai Liu, [3]Wei Lu, [4]Mao Chen
[1,2,3,4]*National Engineering Research Center of E-learning, Central China Normal University, Wuhan 430079, China*
[1, First Author]*Liu.Zheng.jojo@gmail.com,*[2]*L.GuanHai@gmail.com,*[3]*luweilfgl@qq.com,*
[*4]*educhenm@yahoo.com.cn (Corresponding Author)*

Abstract

Currently, there are many two-dimensional Dynamic Geometry System(DGS) which reached a high level of development, while the development of three-dimensional DGS is at its beginning stage and there are only a few 3D DGS existed for solid geometry education, especially in China. These 3D DGS mainly used the plotting module as the core, but it brings some deficiencies such as can't precisely operate and control of figures, geometric and algebraic module can't link closely, the feature of dynamic displaying the changing process is not demonstrate completely. In order to solve these problems and aid the teaching and learning of solid geometry, we proposed and implemented a three-dimensional DGS named Dragon3D Sketchpad with a parameterized model which is the innovation compare to the others three-dimensional DGS. In this paper, firstly, it introduced the architecture and the major components of the system. Then, it described the concept and implementation mechanism of parameterized model. At last, it discussed the advantages of parameterized model with representative examples. The results show that geometric objects could be plotted and manipulated easily in three dimensions, the common functions in a DGS such as animation, transformation, locus and iteration are realized for the spatial objects, and the parameterized model can well solve the problems which mentioning above, it means that the Dragon3D Sketchpad is very useful in solid geometry education.

Keywords *Dynamic Geometry, 3D Dynamic Geometry System, Parameterized Model*

1. Introduction

Geometry, including the plane geometry and solid geometry, is one of the difficult and key content of the mathematics curriculum in the secondary school. In geometry education, interactive geometry software environments such as dynamic geometry system (DGS) have brought great benefits to the teaching and learning of geometry. DGS is a computer application that allows the exact on-screen plotting of geometric constructions, whose typical characteristic is that when some points, lines or polygons are dragged by mouse, other dependent geometrical elements will change accordingly to maintain the geometric properties imposed initially. The dynamic feature revealed in DGS is of great significance in geometry education [1, 2].

Currently, there are dozens of two-dimensional (2D) DGS all over the world, among which The Geometer's Sketchpad [3], Cabri-Geometry [5] and Cinderella [6] are the most popular ones. Compared with the 2D DGS, the development of 3D DGS is much more difficult and complex in the design and implementation. Consequently, there are only a few 3D DGS existed at present, such as Cabri3D [12], Calques3D [13] and Archimedes Geo3D [14]. Moreover, these 3D DGS doesn't fully meet the requirement of the teaching and learning of solid geometry, which mainly manifests in the following aspects: (1) the movement and the change of the objects in the geometric shape are only manipulated by mouse, which could not be precisely enough, and furthermore some critical situation is difficult to be obtained or observed when the shape changes; (2) the change of properties of geometric objects have to be set manually, as a result, some dynamic effect such as the process of iteration changing could not be displayed smoothly and completely. In essence, most of these deficiencies are resulted from the lack of algebra-related module, or the algebra-related module and geometry modules are not linked closely enough in these DGS.

In this paper, we proposed a parameterized model to solve the problem that the algebra related functions are missed in many 3D DGS, and implemented the parameterized model in Dragon3D, a 3D DGS that has been proposed in our previous research [15]. With this model, the movement of geometric objects can be manipulated automatically and precisely, the synchronous changes of several geometry objects could be obtained, and simple algebra calculation could be realized, As a result, both the algebra and geometry are embodied in the DGS and the algebra and geometry are connected closely.

The remainder of the paper is organized as follows: Section 2 presents the architecture of the system; Section 3 details the design and implementation of the parameterized model. Section 4 discusses the advantage of the parameterized model by some examples. Finally, the conclusion and the future work are proposed in Section 5.

2. System architecture with a parameterized model

The system architecture of our DGS is shown in Figure 1, which includes the plotting module, dynamic geometry module, measurement module, geometry transformation module, algebra expression module, and parameterized model module, among which the parameterized model module is at the core of the system architecture.

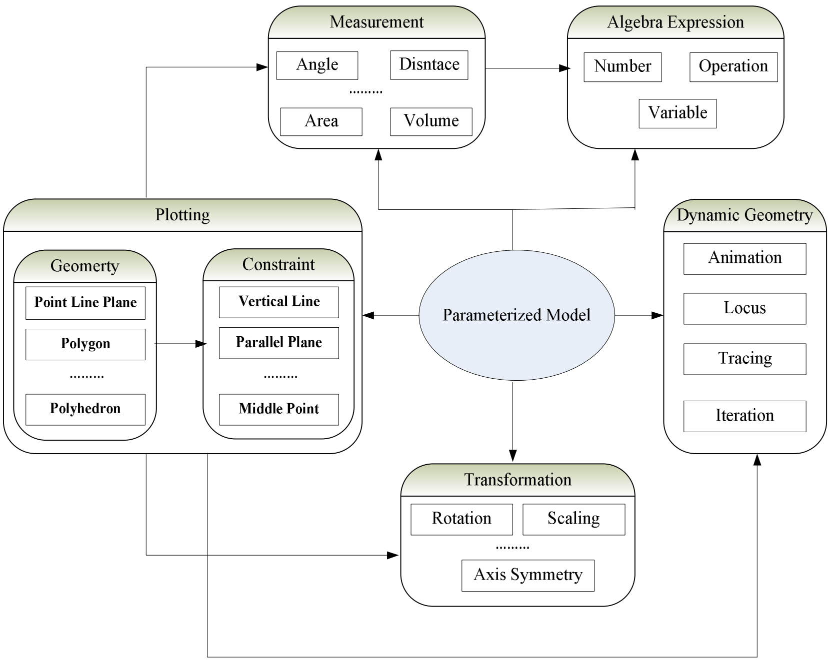


Figure 1. System architecture with the parameterized model

Plotting module could be subdivided into geometry plotting and constraint plotting module. The geometry plotting could be used to plot the commonly used geometry objects such as point(free point, intersection point, point on circle, foot, etc), line(line, segment, ray, vector, etc), circle, conic, triangle, polygon, sphere, cone, cylinder, regular polyhedron, convex, etc, as shown in Figure 2(a), while the constraint plotting is used to determine and maintain the geometric relationships (such as parallel, midpoint, perpendicular, etc) among the geometric elements after plotting, which is one of the key features of DGS. The complex geometric shape in Figure 2(b) is plotted by the constraint plotting.

Dynamic geometry module includes some more advanced functionality such as animation, locus and tracing, which are used to demonstrate different dynamic effects to visualize the abstract geometry properties and spatial concepts. Measurement module could be used to measure the geometry parameters such as volume, area, distance, angle, which provide a good means to demonstrate the invisible geometry parameters, and it also could be used to explore the geometry properties. Transformation module offers the geometry transformation operations to translate, rotate and scale the geometric shapes. Figure 2(c) and (d) show the example of locus and transformation resulted from Dragon3D, respectively.

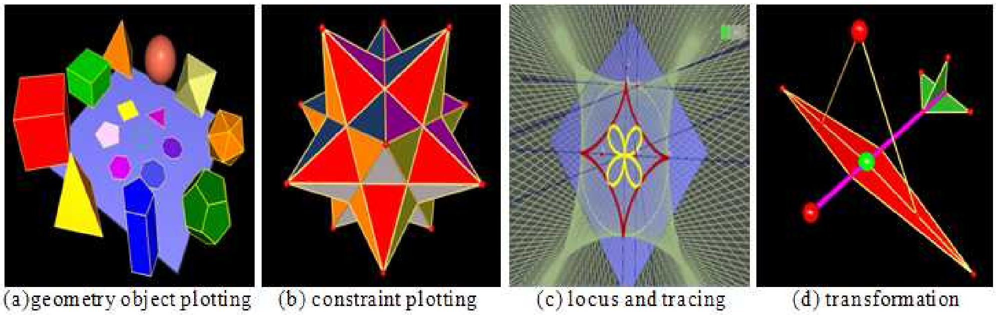


Figure 2. The function demonstration of Dragon3D

Compared with the common 3D DGS such as Cabri3D and our previous version which used plotting module as the core of the system, the parameterized model is at the center of the current system architecture, and it could be looked as a bridge closely connecting other modules. To offer the basic functions of algebraic operations for the parameterized model, an algebra expression module is included in the system, providing the algebraic calculation function for the numbers, variables and algebraic expressions.

3. Parameter and parameterized model

3.1. The idea of combine geometry and algebra by parameter

As we know, geometry and algebra are central to mathematics and have been called its "two formal pillars [16]". Naturally, many geometric shapes such as point, line, circle, plane, polygon, etc, embodied both geometry and algebra, and they can also be regarded as the geometric demonstration of corresponding algebraic knowledge. However, in most DGS, the algebraic connotation of the geometric shapes is missing, and the connection of geometry and algebra are cut off. Accordingly, the algebraic functions are little in the DGS. According to [16], "the soul of mathematics may lie in geometry, but algebra is its heart". Of course, one needs both a heart and a soul. If a DGS system only has geometric function, it would not be able to meet the needs of mathematics teaching and learning. Contrarily, if the algebraic functions are available in the DGS, and the algebraic connotation of geometric objects is embodied, the relationship and interdependence between geometry and algebra would be more easily and vividly understood by the students. So we proposed a idea that using parameter to combine geometry and algebra.

Here take point as an example. Point is the most basic geometric object, based on which other geometric objects could be constructed by it and many dynamic effects could be got by the changing of it. In DGS, point is consisted of several subtypes, such as free point, fixed point, point on line, intersection point and so on. Just as the name implies, free point is the point that can be drag freely with non-constraint, and the fixed point is the point that cannot be dragged by mouse with full-constraint. Actually, point is not only a geometric shape and it also reflects the underlying relationship with algebra. In other words, it could be driven by parameters. For example, a point can be denoted by $p(x, y)$ in plane coordinate system, where x and y are the horizontal and vertical coordinates of it, respectively. If the coordinates of the point are exact value, such as (3, 4), then the point is a fixed point. Furthermore, if the coordinates are $(\cos(t), \sin(t))$, where the functional symbol cos, sin are trigonometric functions and t is a parameter, then the point is a semi-constraint point, which means that the point is on a circle and can only move along the circle with the changing the value of t. When t changes from 0 to 2π, the locus of the point is a circle. Obviously, the locating and moving of the point can all be absolutely controlled by the parameter. Parameter connects the geometry and algebra, and embodies the algebraic connotation of geometric shapes.

With the help of parameter, not only the function of a DGS could be greatly improved, but also the operating would become more convenient and accurate. For instance, in a common DGS, it is almost impossible to move a point that is on a segment to the center position by mouse accurately. However, in the algebraic version, the coordinates of the point could be expressed as a function with parameter t as follow:

$$X(t) = P + t\vec{d} \quad (1)$$

$$\vec{d} = P_1 - P_0 \quad (2)$$

Where P is any point on the segment, the P_0 and P_1 is the begin and end point of the segment , and t is a parameter with range from 0 to 1. If t is set to be 0.5, the point will move to the position of midpoint accurately.

3.2. The mathematical principle of parameter and parameterized model

From the examples above, we can see that the parameter is a variable with a series of properties, such as the label, variable range, current value and step. Many numerical values in DGS can be considered variable and used as a parameter, generally, which could be divided into three categories as follows:

(1) The coordinate of point, which has been shown in the first example above.

(2) Semi-constraint point, the movements of points on line, circle or polygon are semi-free, i.e., they could be moved freely only along the line, circle or polygon, respectively, so they are also called semi-constraint point. The position of semi-constraint point could be parameterized by a variable, just as the second example shown above. In essence, the high-level functions such as animation and locus of DGS are all based on the linear motion of semi-constraint point.

(3) The properties of figures. The size, color, transparency of the geometric shapes, The distance and angle among the geometric shapes, the depth of iteration, the speed of animation and locus, etc., all of them could be parameterized, which could bring great benefit to the dynamic changing ability of the geometric figures.

To cater for the requirement of calculation of parameters and algebra-related teaching and learning, the algebraic calculation function for parameters, variables and algebraic expression is needed, which is the so-called parameterized model in our DGS. With the parameterized model, the movement and changes of geometric shapes could be expressed as a dynamic process driven by parameters as follows.

$$M : X \to m(X);\ X \in F \quad (3)$$

where $X \in F = (x_1, x_2, x_3, x_4 \cdots, x_i)$ is a set of parameters. For each parameter x_i , it could be denoted by the following expression, in which t is the current time and $x(t)$ is the current value of the parameter.

$$x : t \to x(t) \quad (4)$$

4. The implementation of parameterized model

4.1. The mechanism of parameter database

From Figure 3, we can see the mechanism of parameterized model which consist of input module, parameter processor and parameter database. The input module can be further subdivided into three categories which introduced above, and the parameter processor is used to query the parameter whether existed in the database or analysis the expression to find out which parameters the expression contained, parameter database is used to manage the parameters uniformly, the user can use the label which represent the name of the parameter to query the result value in the parameter database.

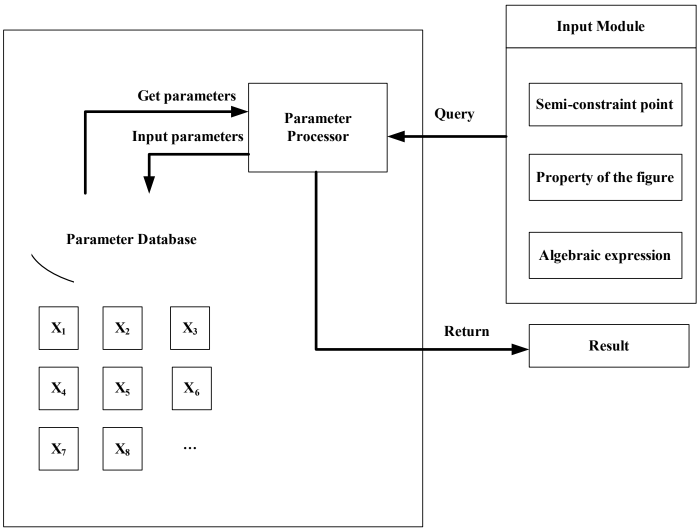


Figure 3. Mechanism of parameter database

We will illustrate the process of parameterization and how the parameter database to deal with the parameters. As mentioned above the user input a expression $f(x) = a * x^2 + b * x + c$, firstly the parameter processor will decompose the expression, then the parameter processor will query the label *a*, *b*, *c* to identify whether it has existed in database, if no input the parameter into the database else get the value of the parameter from the database. At last, the processor will calculate the result of the expression with the parameter value which store in the database. If we keep the parameters *b*, *c* unchanged with number 0 and let parameter *a* changed in the range [-10,10], then tracking the curves of the expression can get the figure 4 from which we can easily observe the influence of the expression curve by the changes of parameter *a*.

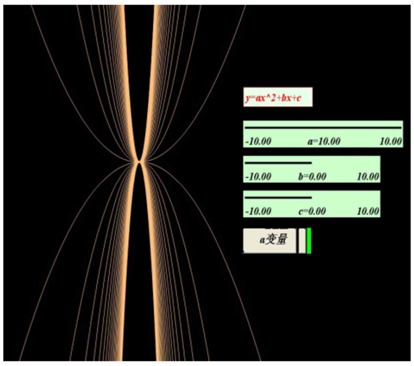


Figure 4. The curves of an expression by changed parameter

4.2. The process of parameter processor to decomposition a expression

Then we will introduce how the parameter processor to deal with the algebraic expression. If the user input an algebraic expression, the parameter processor considers the expression is a composition of a series algebraic items by the operation plus, and every item can be further subdivided into a series factors by the operation multiply. If the factor is still a expression we repeat subdivide it into a series items as above mentioned step until the factor is the basic unit such as a parameter, a number or a common used function like trigonometric function which can be get the value from parameter database or calculated by algebra module, the process of decomposing a expression as shown in Figure 5.

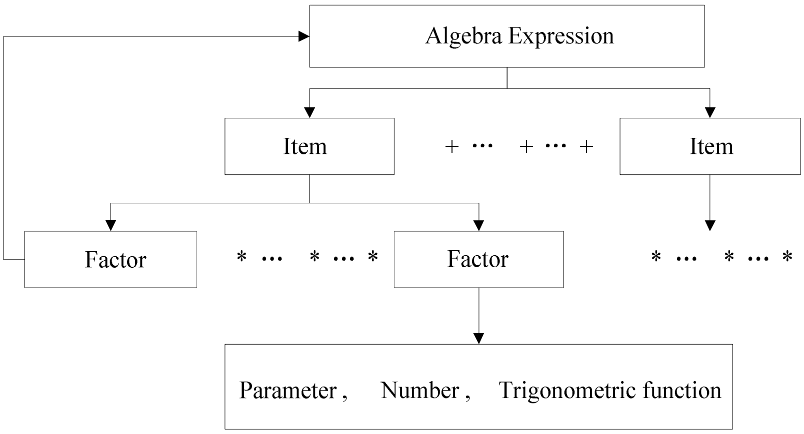


Figure 5. The decomposition process of a algebraic expression

For example, these is a algebraic expression $f(x) = a * x^2 + b * x + c$ with parameters *a*, *b*, *c*, and *x* is the discrete value in the range [-10, 10], the decomposition process is as follows.

- a*x^2+b*x+c
  - a*x^2
    - *a* (Query from parameter database)
    - *
    - x^2
      - x (Discrete value)
      - *
      - x (Discrete value)
  - +
  - b*x
    - *b* (Query from parameter database)
    - *
    - x (Discrete value)
  - +
  - c (Query from parameter database)

Figure 6. The decomposition of a algebraic expression

5. The result of the advantages with parameterized model

Based on the previous version of Dragon3D, the parameterized model and the modification of algebra expression module with the parameterized model are realized. To show the merits of parameterized model, several examples are made by the improved Dragon3D. Generally, four aspects of functions could be obtained with the help of parameterized model, as shown in the following.

5.1 The figure can be more precisely manipulated and controlled

The interaction of figures by graphic interface is intuitive and convenient and the graphic interaction mode of DGS provides user the ability of what you see is what you get. However, as pointed out above, the movement of the figures could not be controlled precisely only by the graphic interaction, as a result, some critical point or value can’t be observed easily when the figures continuous change.

Through the parameterized model, the system can offer a more precise control ability of figures by the setting of parameter with accurate numerical value. For example, if we want to observe the changes in section of a cube and see exactly what the critical value at the cut surface changing, if only by dragging points on the line by mouse, it is difficult to accurately locate these critical points. With the help of parameter of the semi-constraint point, these critical points can be clearly observed, as shown in Figure7.

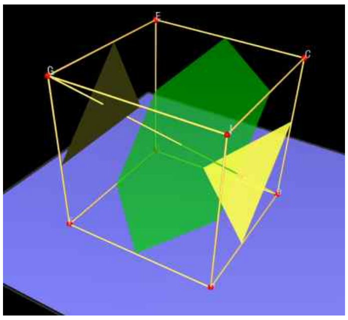
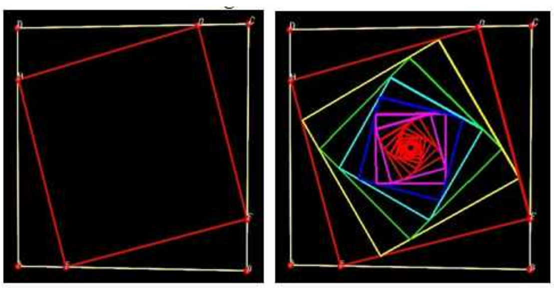


Figure 7. Using parameters to observe the sections of a cube

5.2 Geometry objects or constraints could be linked uniformly

The same parameter can used to connect several geometry objects or constraints and create an internal relation between them, then geometry objects or constraints usually show the same properties because they are connected by the same parameter and controlled by it uniformly.

For example, there is a square *ABCD* with four edges *AB*, *BC*, *CD*, *DA*, and we plot a point *E* on the edge *AB* which with the type PointOnSegment. Since it is a semi-constraint linear point, the system would create a parameter x_1 to control it, and the variable range is from 0 to 1. Then we plot another three points *F*, *G*, *H* on corresponding edges *BC*, *CD*, *DA* with the same parameter x_1 to control them, as shown in the Figure 8(a). After that we used iteration function to map point *A* to *E*, *B* to *F*, *C* to *G*, *D* to *H*, level by level, this process is repeated by function iteration, a beautiful figure is resulted, as shown in Figure 8(b).



(a) (b)

Figure 8. The demonstrate of iteration by semi-constraint points

5.3 Connecting the geometry property and the algebra

As mentioned earlier, the measurement result of geometry property such as length, angles, area and volume can be used as a parameter, and these parameters could be used as common variables in algebraic expressions. As these parameters have geometric meaning, so it is possible to make conjecture about the geometry theorem with them.

For example, if there is a regular tetrahedron and a point inside it, the user could find the sum of the distance from the point to four surfaces is forever equal to the high of the regular tetrahedron. The user could measure the values of four distances and the high which are all geometry properties, then the user can use the algebraic expression to calculate the sum of the four distances and find it forever equal to the high of the regular tetrahedron wherever the point which inside the regular tetrahedron be dragged.

5.4 Giving dynamic changing feature to the figure

The essence of DG is maintaining the geometry constraints while figures change, which is its key difference with common CAD systems, and its emphasis on the presentation and display characteristics of a changing process making it ideal for geometry education.

In order to maximize the feature of dynamic changing, we propose the idea using parameters to set the properties of the figures replace the original numerical value, which can make figures dynamic changing without manually changing the value. For example, the iteration example in Figure 9, if the property is not parameterized, we would have to manually set the value of depth and then observe the changing of the figure, so the continuous and smooth changing process of the figures could not be presented. However if we use parameter to set the property, this shortcoming can be made up easily, this idea can make the discrete changing process become continuous.

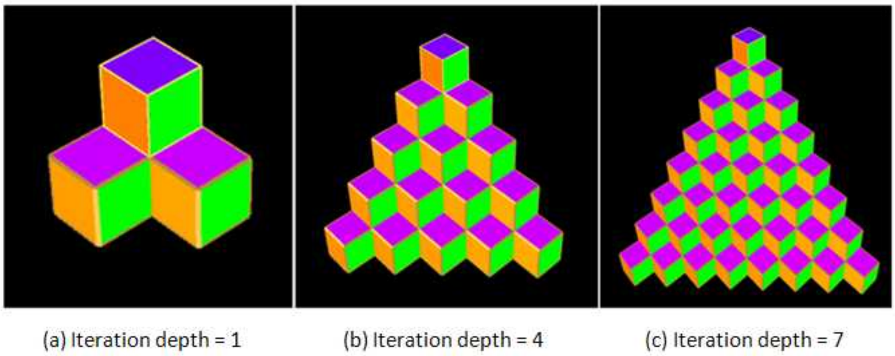


Figure 9. The parameterized of iteration depth

6. Conclusion

In this paper, we presented the design and implementation of a parameterized model for Dragon3D, a 3D DGS for solid geometry education in China. The system architecture of the improved DGS and the implement mechanism of the parameterized model are proposed in details, respectively. To demonstrate the advantages of the parameterized model, several examples are made by the resulted DGS, illustrating its value in geometry education from different aspects. Note that all the examples could not be made by the common 3D DGS such as Cabri3D and the previous version of Dragon3D, where no parameterized model or algebra-related functions available. We hope that through our work in this paper, the idea that “combine number with geometrical figure” could be applied in more DGS.

In the next step, the major work is to research about the automated reasoning of solid geometry and integrate it into Dragon3D, so that it could meet the full range of requirement of teaching and learning of solid geometry.

7. Acknowledgment

This work was supported by National Natural Science Foundation of China (No. 60903023), Specialized Research Fund for the Doctoral Program of Higher Education (No. 200805111011) and self-determined research funds of CCNU from the colleges’ basic research and operation of MOE (No. CCNU11A02009).

7. References

[1] K. Ruthven, S. Hennessy, “Constructions of dynamic geometry: A study of the interpretative flexibility of educational system in classroom practice”, Computers and Education, vol. 51, no. 1, pp. 297-317, 2008.
[2] B. Ubuz, I. Üsün, A. K. Ebraç, “Effect of dynamic geometry environment on immediate and retention level achievements of seventh grade students”, Eurasian Journal of Educational Research, vol.35, pp. 147-164, 2009.
[3] N. Jackiw, The Geometer's Sketchpad, CA: Key Curriculum Press, 1995
[4] J. Z. Zhang, C. Z. Li, “An Introduction to Logical Animation”, In Proceeding of the 6th Computer Algebra and Geometric Algebra with Applications, pp. 418-428, 2004.
[5] X. S. Gao, C. C. Zhu, S. C. Chou, Geometry expert, Taiwan: Nine Chapters Publishers, 1998
[6] U. Kortenkamp, Foundations of Dynamic Geometry, Ph.d Thesis, Swiss Federal Institute of Technology Zurich, 1999.
[7] F. Botana, J. L. Valcarce, “A dynamic-symbolic interface for geometric theorem discovery”, Computer and Education, vol. 38, no. 3, pp. 21-25, 2002.
[8] A. Gutierrez, “Visualization in 3-dimensional geometry: In search of a framework”, In Proceeding of the 20th conference of the international group for the psychology of mathematics education, pp. 3-19, 1992.
[9] M. Battista, “On Greeno's Environmental/Model View of Conceptual Domains: A Spatial/Geometric Perspective”, Journal for Research in Mathematics Education, vol. 25, no. 1, pp. 86-94, 1994.
[10] H. Hauptman, “Enhancement of spatial thinking with Virtual Spaces 1.0”, Computer and Education, vol. 54, no. 1, pp. 123-135, 2010.
[11] S. Isotani, L. O. Brandã, “An algorithm for automatic checking of exercises in a dynamic geometry system: iGeom”, Computer and Education, vol. 51, no. 3, pp. 1283-1303, 2008.
[12] E. Bainville, J. M. Laborde, Cabri3D v.1.0.3, Cabrilog, http://www.cabri.com
[13] N. V. Labeke, Caqules3D, http://www.calques3d.org
[14] A. Goebel, Archimedes Geo3D, http://raumgeometrie.de/drupal/
[15] Z. Liu, G. H. Liu, W. Lu, M. Chen, “Design and Implementation of a 3D Dynamic Geometry System”, AISS: Advances in information Sciences and Service Sciences, 2012(in press).
[16] M. Atiyah, “Mathematics in the 20th Century”, American Mathematical Monthly, vol. 108, no.7, pp. 654-666, 2001.
[17] M. H. Su, P. T. Yu, “A Directable and Designable Course Recording System”, JCIT: Journal of Convergence Information Technology, vol. 6, no. 3, pp. 234-243, 2011.