

Reproduction of Chemical Experiment Based on Computer 2D Animation Technology

Yanqing Gao

Software College, Nanyang Institute of Technology, Nanyang, Henan 473004, China
 yanqing@zzu.edu.cn

In order to solve problems such as operation and control in the process of chemical experiment in teaching, using the technology of digitized computer at present, the operation steps, materials, speed, control and reaction of chemical experiments can be directly and accurately displayed on the video screen in the form of two-dimensional animation. This technology has made it skillfully restore the chemical experiment scenes in the training room, reduce the labor intensity of the experimenters, and improve the accuracy of the chemical experiment, hoping to make certain contributions to the research of chemical experiments. In this paper, the shortcomings of traditional chemical experiments are compared. The design of computer animation and the implementation of chemical experiments are expounded, based on vector graphics and made by Flash MX software. It is feasible to use computer two-dimensional animation to show chemical experiments. It has revolutionized the methods of chemical experimental research, made the research of chemical experiments more digitalized, and is worth popularizing.

1. Introduction

The computer 2D animation rendering technology displays the chemical experiment process on the video screen in order to better present its integrity (Tsutsuguchi et al., 2015). 2D animation technology concatenates the images on the plane with an effect from stationary to motive, so that the real 3D space simulation is thereby achieved. Any way you look at 2D screen, the content remains constant. The 2D animation technology can simulate chemical experiments of several types, such as replication, design and comprehensive experiments. In general, these different types of experiments are processed by using Ulead GIF-Animator, Adobe PS and Flash. The 2D computer animation restores chemical experiments more intuitively, visually and with more informative content (Zhang, 2015).

2. Design of 2D computer animation and implementation of chemical experiments

The 2D animations are produced by FlashMX based on vector graphics, and its simulation can be controlled precisely by the ActionScript language provided with FlashMX. The FlashMX designs lively and high-definition 2D animation without programming, but the 2D animation files produced by it has a tiny capacity and easy to learn. These are very popular among the broad masses of animators (Hamdan and Ali, 2015). This paper adopts FlashMX for 2D animation by which the demos for chemical experiment instruction can be enabled, such as institutive demos of the experimental device, process and results with 2D animation, combined with audios, videos and images. The schematic diagram of reproduction of chemical experiments with 2D animation is shown in Figure 1:

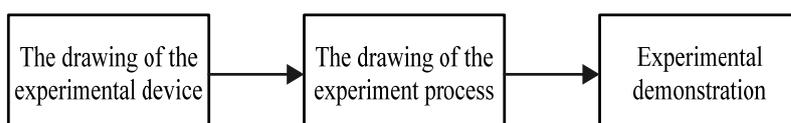


Figure 1: Experiment flow chart

2.1 Rendering of the experimental device

The AutoCAD renders the experimental devices and instruments which are imported into FlashMX after being transformed into appropriate formats. The pipelines, devices and instruments are connected one by one to ensure the integrity of the experimental equipment. At last, the experiment flow chart is rendered.

2.2 Rendering of experiment workflow

After the flow chart of experiment device is rendered, 2D animation is used with audios, videos and images to illustrate the name of the experiment, experiment device and instrument, reagent and dosage, the reaction status, the experimental steps are indicated by the arrows; the background music and the character images are extracted by FlashMX to achieve a real effect, and finally generate the appropriate playback file format.

2.3 Experiment demo

A ready-made 2D animation file is inserted into the digital multimedia equipment in the classroom, play back the relevant chemical experiments, from the experiment equipment to the experiment operation, and finally to the experimental treatment, so as to get a live show on the video screen of projector.

Here is an example, i.e. Reynolds experiment, about how a simple 2D computer animation reproduces the chemical experiment. In 1883, Reynolds, a British scientist, revealed the fluid flow mechanism and found that under the conditions of laminar flow and turbulence, the fluid flow shows different phenomena. A proportional relationship between head loss and flow velocity was proposed (Galvanin et al, 2015). The physical experiment principle is to observe how the internal texture moves when the fluid flows in the experiment device and what is the impact of various factors on fluid flow. These 2D computer animation technology reappear the principle of chemical experiments. The 2D animation is produced in the following ways:

- (1) A file named Renault experiment is created with FlashMX, adjust its animation pixel to 650 * 600, and select the white background, etc. as the initial settings.
- (2) The rendering area is adjusted to the grid form of ruler, and render it in the scenario window.
- (3) Some relevant tools in FlashMX are used to render the Renault experiment devices, the experiment flow chart and reaction diagram, etc., concatenated with the well-established animation components into a continuous screen effect, that is, 2D animation. The experiment is thus presented again in the form of video (Nakagawa, 2014).

3. Simulation realization of computer graphics classical algorithm based on Flash MX

In recent years, the advent of multimedia technology and CAI has diversified the forms of learning and make it lively and vivid, coupled with a variety of teaching coursewares sprung up. However, there is almost no change in the Exposition part involving the basic algorithms in today's graphics courseware (Kim et al., 2015). This paper proposes a graphics algorithm simulation demo method which can process any image input by users and auto generate demo content, enable sync display of intermediate processes performed by algorithm and processing results of input images, free adjustment of the program execution speed even single-step execution, so that learners are fully in the initiative status. It features a good interaction.

3.1 Implementation principle of algorithm simulation demo

Any algorithm includes a series of steps, each of which takes appropriate intermediate data and results as the key to correct understanding of the algorithm. Any of algorithms can eventually be implemented in a program language, but cannot output final results until the full code is executed. Many crucial intermediate results will be unable to be displayed. Even though these codes are executed by a single step, the existing programming environment can only display the results of intermediate procedures with abstract characters, numbers. If the invisible data processing is expressed as intuitive changes in graphics, this will inevitably make us an inedible impression on the algorithms, and much easier to grasp and apply them. The database structure is built as shown in Table 1.

Whether in the structured programming language or in the object-oriented programming language programming, the basic statements are nothing more than Order, If, and Loop. Flash MX animation consists of frames played by the main timeline, and perform ActionScript on each keyframe during playback. If the algorithm is written in turn on each keyframe step by step, it is equivalent to a sequential execution of the algorithm program when playing the animation. The combination of ActionScript flow control command if ... else and the animation control command gotoAndPlay () implements the functions of If and Loop statements. Two commonly used basic codes are given as follows, but the actual application will vary with the specific criteria.

Table 1: Establishment of DB structure

Name	Type	Explain
Classification number	Text	The classification number of experimental materials
Name	Text	The name of the experimental material
Category	Text	The category of experimental materials
Carrier form	Text	The carrier form of the experimental material

3.1.1 If statement code:

```
if (the condition is true)
{gotoAndPlay (frame A);}
else
{gotoAndPlay (frame B);}
```

3.1.2 Loop statement code:

```
if (the condition is true)
{gotoAndPlay (a frame precedes current one);}
```

That Flash MX control commands match frames can simulate the execution of actual program, which is the basis for implementing algorithm simulation demo.

3.2 Concrete implementation of algorithm simulation demo

Each demo consists of 4 basic modules: i.e. graphics input, graphics processing, intermediate process display, and control program execution. Learner first renders the images required to be processed, such as line segments, polygons, etc., with the graphic input module by using the mouse; The control program execution module performs the program in a continuous or single step mode according to the learner's requirement. The graphics processing module deals with the appropriate algorithm for input graphics. The intermediate data processed each step will be displayed on appropriate module in this process, while reflecting processed results of current graphics. Its logical structure is shown in Figure 2:

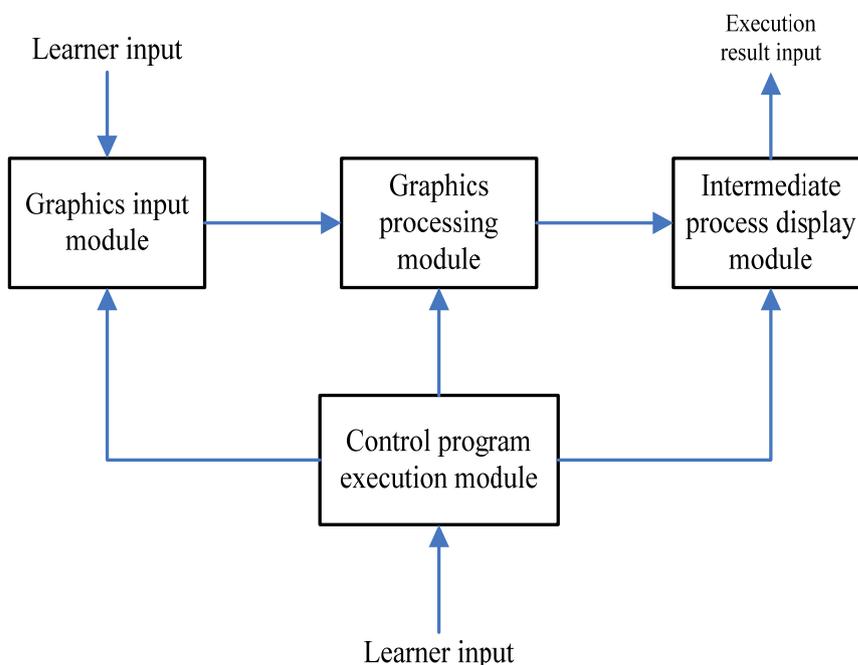


Figure 2: Logical structure of algorithm simulation

The intimate cooperation of 4 basic modules makes up simulation demos for each algorithm, any of which are indispensable. The following sections focus more on the specific implementation of 4 modules.

3.2.1 Graphic input module

The module enables interactive input of graphics, learners use the mouse enter lines, polygons, curves and so on in the Graphic Input window at the request of the algorithm. If the input graphs are undesirable, you can also delete it and re-enter others, so that the module can render and modify the basic graphics. The event manipulation functions `onClipEvent(mouseDown)`, `onClipEvent(mouseMove)`, `onClipEvent(mouseUp)` in Flash MX, used with the drawing function, can achieve the input of basic graphics. Some functions for drawing a straight line is shown as follows:

```
Function drawLine(x1, y1, x2, y2)
{duplicateMovieClip ("originalLine", "line", 1);
line._x = x1;
line._y = y1;
line.xscale = x2-x1;
line.yscale = y2-y1;}
```

Where `originalLine` is a `MovieClip` instance, composed of a line segment from points (0, 0) to (100, 100). If the `originalLine` is changed to a circle with a center of (50, 50), a diameter of 100, or a square with a top left vertex at (0, 0) and a bottom right vertex at (100, 100), such function can be used to draw an oval, circle or rectangle, square. When creating this symbol, the `linetype` must use the `hairline`, otherwise the resultant graphics will be uneven in thickness due to their different sizes. The line segment underlies the polygon input as a continuous input of the line segment. What we should note is that when replicating the above line segment with `duplicateMovieClip`, you must assign a unique depth level to `MovieClip` to prevent it from replacing the existing line segment with equal depth, as shown in Figure 3.

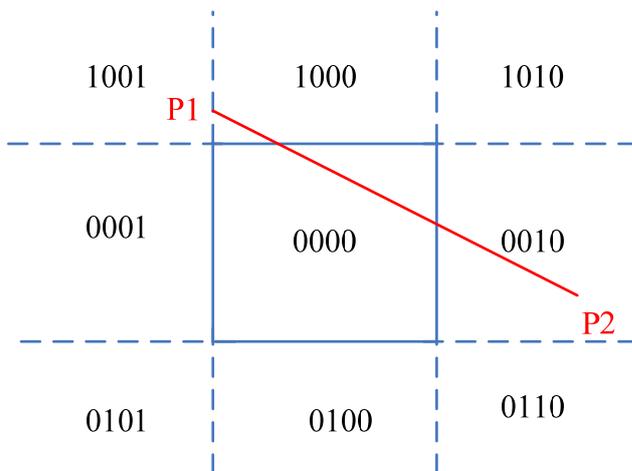


Figure 3: Line segment cut arbitrarily

3.2.2 Graphics manipulation module

After learner inputs the graphics (line segments, polygons, etc.) to be processed, the graphics input module has stored data about the graphics as inputs of the graphics manipulation module which, in turn, processes the input graphics with different algorithms to get the results of their respective algorithms implemented in ActionScript.

3.2.3 Intermediate process display module

This module will display data input by the graphics manipulation module in the form of the images or texts. At the time when the input graphics are processed, the results should be timely reflected on the graphics, and enables a sync display about which procedures the current program has executed, which can be reflected with flow chart, pseudocode or algorithmic text, in the light of specific criteria. Not only that, the dynamic text box is also used to display some of key data, for example, the Cohen-Sutherland clipping algorithm for demos of line

segment shows the program flow chart of the algorithm. When tailoring a line segment drawn arbitrarily by learners, the upper left window displays the intersections of this line with left, right, lower and upper borders of clipping window one after another and the rest parts after it is tailored. The dynamic text box on the right window shows the codes of two ends of current line segment, and while in the program flow chart, reversely displays which procedures the current program runs, as shown in Figure 4.

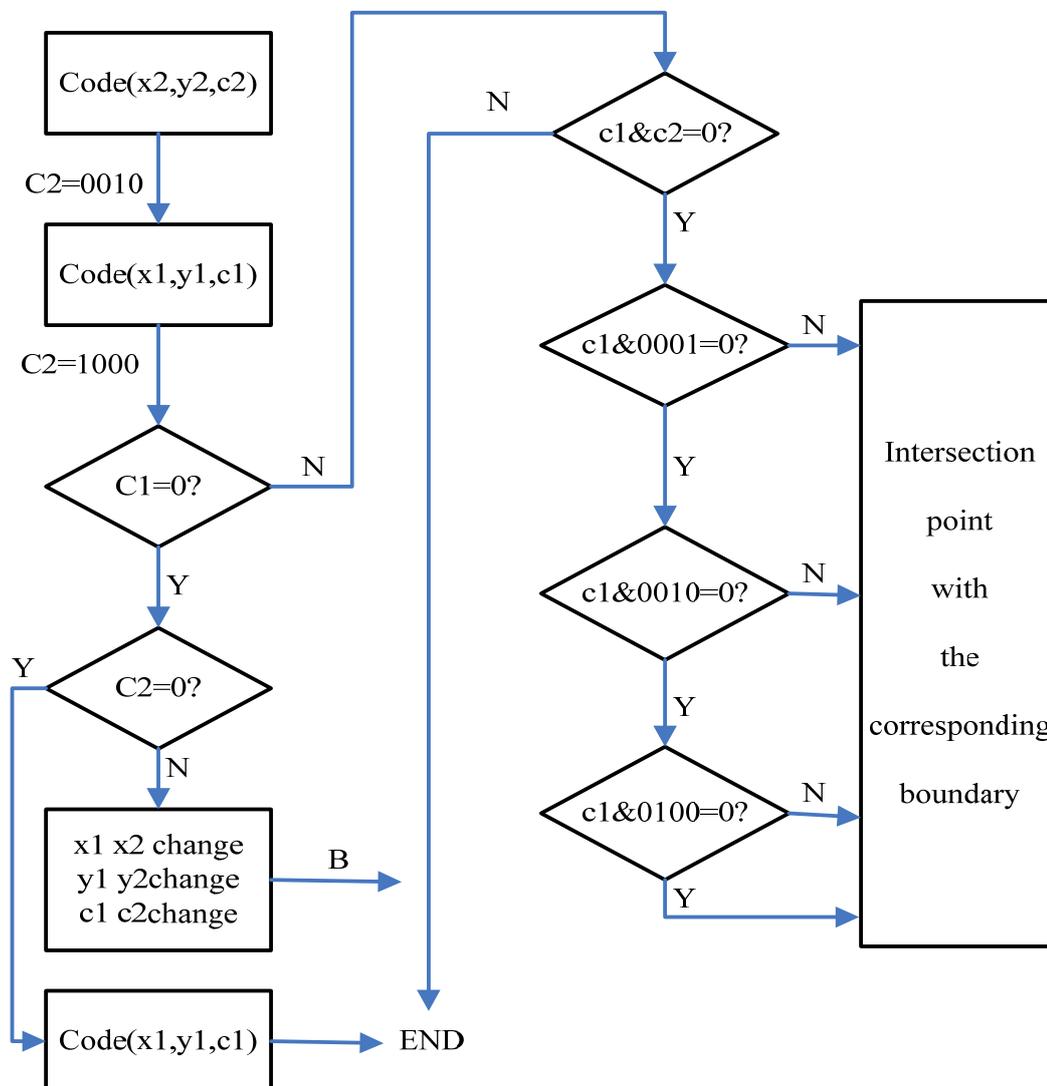


Figure 4: Demo of clipping algorithm for line segments

4. Conclusion

In short, the essence of using computer two-dimensional animation technology to reproduce chemical experiments is to transform specific human operation experiment preparation, experimental process, experimental results into two-dimensional animation virtual form, so that the experimenter is more concise and intuitive to discover and study related chemical experiments. It breaks away from the traditional explanation and experiment, and reduces the number of experiments repeatedly. The experimenters can observe the whole experiment process intuitively through computer 2D animation, and speculate on the relevant experiments based on this experiment, and then prove it again. According to the above, it is feasible to use computer 2D animation to show chemical experiments. It has revolutionized the methods of chemical experiment research, made the research of chemical experiments more digitalized, and is worth popularizing.

Reference

- Fournier A., Fussell D., Carpenter L., 2015, Computer rendering of stochastic models, *Communications of the Acm*, 25(6), 371-384, DOI: 10.1145/358523.358553
- Galvanin F., Cao E., Al-Rifai N., 2015, Model-based design of experiments for the identification of kinetic models in microreactor platforms, *Computer Aided Chemical Engineering*, 37, 323-328, DOI: 10.1016/B978-0-444-63578-5.50049-9
- Hamdan M.N., Ali A.Z.M., 2015, Developing and evaluating of non-realistic three-dimensional (3d-nr) and two-dimensional (2d) talking-head animation courseware, *Malaysian Online Journal of Educational Technology*, 3.
- Kim S., Guy S.J., Hillesland K., 2015, Velocity-based modeling of physical interactions in dense crowds, *Visual Computer*, 31(5), 541-555, DOI: 10.1007/s00371-014-0946-1
- Konicki M.G., Adeli H., 2015, A three-dimensional animation system for seismic response of multistory structures, *International Journal of Imaging Systems & Technology*, 8(3), 313-321, DOI: 10.1002/(SICI)1098-1098(1997)8:3.
- Nakagawa H., 2014, Demonstration of reappearance in the Long Period Ground Motion Using the Two Dimensional Earthquake Shaking Simulator, *Summaries of technical papers of annual meeting*, 915-916, Architectural Institute of Japan.
- Tsutsuguchi K., Shimada S., Suenaga Y., 2015, Human walking animation based on foot reaction force in the three - dimensional virtual world, *Computer Animation & Virtual Worlds*, 11(1), 3-16, DOI: 10.1002/(SICI)1099-1778(200002)11:1.
- Zhang R., 2015, Inspiration of Traditional Ash-Bashing Pictures to Image Design of Two-Dimensional Animation, *International Conference on Information Science and Control Engineering*, 269-273, IEEE, DOI: 10.1109/ICISCE.2015.66.