# A Study on Data Structures to Draw Figures Efficiently

Takashi Amano

School of Information Science,
Japan Advanced Institute of Science and Technology

February 15, 2001

**Keywords:** group updates, data structure, chromatic tree, thinking time.

When we solve a problem treating geometric data, the role played by computers is very large and the necessity of dealing with the problem concerning geometric figures efficiently by computers has increased more and more in recent years. Studies on data structures to manage geometric data are performed rigorously, and various data sutructures such as an AVL tree and a segment tree are proposed.

Drawing tools, such as CAD, must be able to insert, delete and select required geometric objects quickly while they treat geometric huge data. For that, we need suitable algorithms and data structures. Moreover, it is important that a user does not feel unconfortable with the processing time. Now, generally we use balanced binary search trees represented by AVL tree for basic data structures in CAD. On an AVL tree, a data structure is updated for every insertion or deletion. Geometric data are often huge. Thus, when the updating is performed frequently, processing takes much time and the problem that a response may be overdue arises. Therefore, we also need a technique for collecting many updates into a group and performing insertions and deletions at once. The technique of collecting at once and updating is called "group updates".

In this study, we consider effective use of thinking time (input waiting time) made when an operator are drawing figures. We have developed an algorithm and data structure for practical use so that selection, insertion and deletion are done quickly using group updates.

When an operator draws figures, generally no one inputs continuously, but does after considering figures to draw. A computer enters a state for waiting an input in which no figure is input. That is, drawing time can be divided into time for actually input of figures and thinking time of figures to draw. An operator draws figures by repeating the two processes. We use effectively the waiting time of computers created by the characteristic of the input by human operators. If a computer does some updates during waiting time,

we can perform operations of drawing smoothly, and we expect that responses can also be performed quickly. Although a computer does updates during thinking time, a computer must perform indispensable processings immediately. For an oparator doesn't understand anymore what figures are drawn where. In this study, when an operator inputs, a computer performs indispensable processings immediately, and we try to draw figures efficiently by updating the data structure which needs time later.

We prepare three memory domains to accumulate data as data structures for the technique. The main memory domain accumulate all data, and others are memory domains to accumulate data temporarily. They are a memory domain which accumulates data to be inserted, and a memory domain which accumulates data to be deleted. The data to be inserted when an operator draws figures, or data to be deleted are saved to the memory domain accumulated temporarily. Next, during thinking time, the data accumulated temporarily until now are inserted in the main memory domain or data are deleted from the main memory domain. Since all data are accumulated there when the number of memory domains is one, the conventional technique requires time for updates. Since the technique in this study accumulates only the present updates temporarily and updates for the main are performed later, time required for the present updates is shortened.

The updating operations accumulated temporarily are not reflected in the main data structure by the simple technique, but we elaborate a device on group updates so that it can update more efficiently than a simple technique. A simple technique inserts into or deletes from the main memory domain for every data. We use a chromatic tree as a data structure required to perform group updates efficiently.

In this study, we experimented by programming the tool which performs group updates using a chromatic tree. We obtained the results that the group updates can perform more quickly than the conventional technique by the experiments of comparing the conventional techniques with group updates in execution time. From the experimental results, it is seen that the technique of this study can be updated more efficiently than the conventional technique and it was shown that drawing of figures can also be performed smoothly.

Since updating operation does not update the main data structure unlike the conventional technique, we can perform input operations smoothly. Moreover, in group updates, since it is divided into the main data structure and the temporary memory domains, when a certain data are input and operations of deletion are performed immediately, there is an advantage that offset of the data in a temporary memory domains can be performed quickly.

Although we have realized a merit of a group updates using a chromatic tree, we could use a skip list. In the future, we would like to use a skip list instead of a chromatic tree. Then, as compared with the technique proposed in this study, we will obtain an interesting result. Since a chromatic tree is a data structure also efficient for parallel proceccing, we will also need to program and compare the data structure to enable parallel proceccing. In this data structure we have only dealt with segments. However, it became easier to draw figures if the elements which form figures have basic elements, such as not only segments but also points, circles and polygons, and curves. Study on data structures for accumulating such basic elements efficiently has been left as a future subject.