Image Representation Using Finite State Automata

Charles Quenneville et Jean Meunier
Département d'informatique et de recherche opérationnelle,
Université de Montréal, Montréal, Québec Canada H3C 3J7
514-343-7107
e-mail address : meunier@iro.umontreal.ca

ABSTRACT

Image representation using quadtrees is useful in many applications and is well known in computer graphics. It has been shown that we can represent quadtrees as rational languages that are recognizable by finite state automata (FSA) [1]. These FSA can use less memory and are more flexible than their corresponding quadtrees. Binary image representation using FSA follows these two main steps: (1) FSA construction based on quadtrees and (2) FSA minimization [2]. This approach can be easily adapted to represent gray scale images, colors images or volume data (octrees). Image representation using FSA supports all quadtree operations (union, intersection, rotations, ...) as well as all FSA-defined operations.

KEYWORDS: image representation, automaton, quadtrees.

INTRODUCTION

The weakness of quadtrees appears for patterned images, because those quadtrees contain some redundancy; this redundancy can be minimized by using FSA [2]:

$$M = <Q, \Sigma, \delta, q0, F>$$

where M is the FSA, Q the set of states that forms the FSA, Σ the set of symbols that the FSA accepts, δ the transition function from a state of the FSA to another state according to a symbol of Σ, q0 the initial state of the FSA and F the set of final or accepting states. In our application, the set Σ is {0,1,2,3}, each symbol corresponding to a label of a quadrant of the image.
To construct the FSA from an image, we follow two steps: FSA construction and FSA minimization.

FSA CONSTRUCTION

The construction of an FSA uses any quadtree construction algorithm, but instead of adding a branch to the quadtree, if a quadrant of the image is not of the same color, we add a transition in the FSA on the symbol corresponding to the label of that quadrant; the branches to the leaves corresponds to the transitions to the final state of the FSA (fig 1). The set F of an FSA that represents binary images is a singleton. For gray scale images, the cardinality of that set is the number of gray levels. To represent color images, we can use three FSA, one for each RGB component. Finally, for volume data, the set Σ becomes {0,1,2,3,4,5,6,7} and represents each octant of the octree.

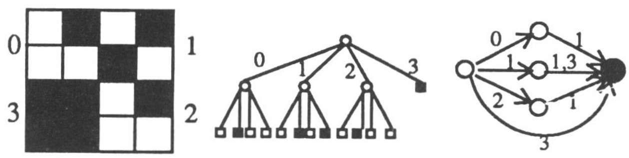

Figure 1. An image, its quadtree and its FSA.

FSA MINIMIZATION

The minimization algorithm is unique to FSA. It takes the redundancy in the FSA and, based on the equivalence between states, merges states that yield an equivalent transition on the same symbols. The result is an FSA with a number of states smaller or equal to the FSA before minimization. In figure 1, the FSA is minimized to three states.

RESULTS

Here are some results for binary images (fig 2):

	Nb. states before min	Nb. states after min.	Nb. nodes in quadtree
1.Mona Lisa	1544	552	3836
2.chess board	87382	10	218453
3.finger print	9533	1936	23869

Table 1: Coding results with FSA and quadtrees.

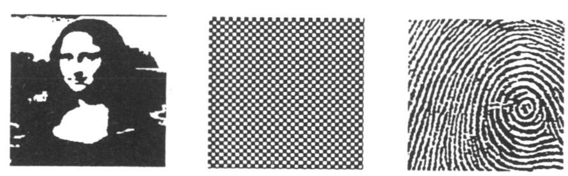

Figure 2. Coded images.

CONCLUSION

We have shown the potential of FSA to represent images and volume data with a better efficiency than quadtrees or octrees and with full support of all image operations [3].

REFERENCES

1. Berstel, J. Compact Representation of Patterns by Finite Automata, Comm. Pixim '89, Paris, septembre 89.
2. Hopcroft, J. Introduction to Automata Theory, Languages and Computation, Addison-Wesley, 1979.
3. Quenneville, C. Memoire de Maîtrise, U. de M. 1993.