On the Complexity of Image Processing and Pattern Recognition Algorithms

Maxim Makatchev† and Sherman Y. T. Lang‡

Dept. of Manufacturing Engineering and Engineering Management,
City University of Hong Kong
† memaxim@cityu.edu.hk, ‡ melang@cityu.edu.hk

Abstract

We study the complexity of image processing and pattern recognition (IPPR) algorithms by their representation as finite cellular automata-based structures. A universal model to represent multilayer homogeneous IPPR algorithms and a technique to compare their quality are required for problems of vision system adaptation, learning, and by the systems for automatic programming. We propose the finite cellular automata-based model for representation of IPPR algorithms and sequential and parallel time complexity measures for this model. Composition and decomposition transformations of proposed structure are suggested and we show that in particular cases they can lead to reduction of complexity. Specific properties of IPPR tasks that are important for their complexity research are discussed.

1 Introduction

The inherent features of many IPPR algorithms include uniform processing over a pattern by the mean of a local mapping from one homogeneous layer no another (so called *mesh-local* algorithms, e. g. some filtering algorithms [13] and full search block matching [7]). The relation of these features with formal cellular automata (CA) inspired the appearance of a number of CA-based models for the implementations of IPPR algorithms (e. g. [1], [2], [3], [6], [9], [11], [12], [14]), of IPPR-related data compression and encryption algorithms [10]. CA-based models are also utilized for a design of VLSI circuits(e. g. [8], [15], [19]) and cellular structures with local connections are a well known parallel multiprocessor architecture, referred to as a *mesh* (e. g. [16]). The importance of applications and variety of different but related models of CA-based architectures led to generalizations utilized for investigation of computational power of particular models as well as of a general architecture itself [4], [5].

In this paper we consider the problem of pattern recognition (or generally IPPR problem) for configurations of cellular lattice of bounded size. In practice such problems are usually formulated by some description of local maps for every layer of processing. To solve the problem one is required only to specify this description for a particular implementation of the algorithm. Thus the operator corresponding to IPPR task is defined constructively. This approach has two major difficulties. Firstly, it is usually not easy to obtain such a definition of IPPR operator, as a deep understanding of the structure of IPPR problem is required. Secondly, once we are fortunate enough to obtain such a definition it is not necessarily optimal. The IPPR problem may still have some concealed mesh-local features that would allow reduction of the complexity of solution. As derivation of an operator usually requires an expert, these difficulties are particularly significant in systems of automatic programming and learning.

One can possibly generalize the discussed

problems into the following: given *some* description of an operator to reveal its mesh-local features, design a "good" mesh-local algorithm as an effective implementation of this operator. A possible feasible description of an IPPR operator is its selected pairs of input and output configurations. Thus the problem can be reformulated as an interpolation of this operator by an effective mesh-local algorithm. Another type of description is a mesh-local (or possibly some other) algorithm corresponding to a particular IPPR operator. In this case the problem is to optimize this algorithm (for example, by a number of iterations).

In our paper we consider the latter case, introducing operations over the structure of a mesh-local algorithm that can be used for its optimization. Namely for a given mesh-local algorithm we define the operation of the addition of the layer, i. e. *decomposition* of the layer, and the reversed operation of the subtraction of the layer, i. e. *composition* of two layers. Using IPPR task with mesh-local properties of different scales, we show that decomposition can lead to the significant optimization. Similarly the possibility of optimization by composition of two layers is shown.

In the next section we provide necessary definitions of the CA-based model utilized for representation of mesh-local algorithms. Section §2.2 defines complexity measures for this model to some extent reflecting the peculiarities of the complexity of realization of local maps. In §2.3, we define two operations over the proposed model for mesh-local computations. Section §3 provides results on the possibility of optimization of mesh-local IPPR algorithms. In §4, we conclude and observe possible applications of the proposed techniques.

2 Definitions

2.1 Finite dynamic cellular automata

We define a *finite dynamic cellular automaton (or FDCA)* as an infinite set of *layers* σ_i, $i = 0, 1, \dots$, where $\sigma_i = (d, k_i, N_i, f_i)$. Here d denotes the dimension of the grid of cells $C = [m_1] \times [m_2] \dots \times [m_d]$ and $[m]$ denotes the set $\{1, 2, \dots, m\}$. Every cell of the layer i can assume states from a finite set $\Sigma_i = 0, 1, \dots, k_i - 1$, $k_i \geq 2$ of possible states that is called the *alphabet* of the layer i automaton. The function $f_i : \Sigma_{i-1}^{N_i} \to \Sigma_i$ is the *local map* of the layer i automaton. For the sake of simplicity we assume that N_i the *basic neighborhood* of the map of layer i, is of form $N_i := [-r_{i,1}, r_{i,1}] \times [-r_{i,2}, r_{i,2}] \times \dots \times [-r_{i,d}, r_{i,d}] \subseteq \mathbf{Z}^d$, where $r_{i,j} \geq 1, j = 1, \dots, d$. Local map and basic neighborhood of layer 0 are not defined.

We call a map $X_i : C \to \Sigma_i$ from the set C of all cells to the layer i alphabet a *configuration* of the layer i of the dynamic cellular automaton and $\mathbf{X_i}$ denotes the space of all possible configurations of the layer i. The layer i local map f_i induces a layer i *global map* $F_i : \mathbf{X_{i-1}} \to \mathbf{X_i}$. For particular configuration X_i and a cell $c \in C$ we define the *local configuration* at c, $x_{i,c} =: N_{i+1} \to \Sigma_i$, by $x_{i,c}(z) := X_i(c+z)$. Now we can define the layer i global map by $F_i(X_{i-1})(c) := f_i(x_{i-1,c})$.

Using both finite cellular automata and finite dynamic cellular automata we face the following technical difficulty: the local rule cannot be applied to cells close to boundary of the automaton. There are known a few possible solutions to this problem for finite cellular automata. One of the solutions is to use *cyclic boundary conditions*, i. e. glue together opposite ends in the grid. Another solution is to adopt *fixed boundary conditions* and assume that the missing cells are in some special state. In this paper we use fixed boundary conditions to solve this problem.

Thus a given FDCA at its particular layer of automaton i computes an operator from a set of configurations of layer 0 into set of configurations of layer $T - 1$, say $P : \mathbf{X_0} \to \mathbf{X_{T-1}}$. The general IPPR task can now be formulated as an operator from one set of configurations to another and the problem is to design FDCA that computes this operator at some layer of its automaton. Thus in this paper we consider only a finite number of first layers of FDCA and the operator corresponding to it. The quality of an implementation can be derived using the complexity measures for FDCA.

2.2 Complexity of FDCA

Now we give the general definitions of two proposed complexity measures for FDCA.

We define the *sequential time complexity* of FDCA consisting of T layers $0, \ldots, T-1$ as the sum of sequential time complexities of all samples of the local maps (one per cell) for all layers, i.e. $L(\sigma) = |C| \sum_{i=1}^{T-1} L_i$, where L_i is the sequential time complexity of the local map of layer i. Further the product $|C|L_i$ is referred to as the *sequential time complexity of layer* i. Similarly we define the *parallel time complexity* of FDCA consisting of T layers as the sum of the parallel time complexities of all layers, i. e. $LT(\sigma) = \sum_{i=1}^{T-1} LT_i$, where LT_i is the *parallel time complexity of layer* i that we set equal to the parallel time complexity of the local map of layer i, since we assume all mappings of the same layer run simultaneously. By definition, if the sequential time complexity of a local map is equal to its space (size) complexity, the sequential time complexity of FDCA corresponds to the space (size) complexity of FDCA.

Sequential and parallel time complexities of the local map $f_i : \Sigma_{i-1}^{N_i} \to \Sigma_i$ of layer i may be defined according to a particular task. In the rest of this paper we assume that a local map is computed by a sequential algorithm, thus its sequential and parallel time complexities are equal and often are referred to as the complexity of the local map.

2.3 Operations over FDCA

Here we introduce two types of transformations over FDCA that are invariant with respect to operator corresponding to it.

Consider the global map $F_i : \mathbf{X_{i-1}} \to \mathbf{X_i}$ from layer $i-1$ to layer i. Assume F_i may be decompose into maps F'' and F' such that $F_i = F'' \circ F'$, by addition of the new layer σ' between layers $i-1$ and i, so $F' : \mathbf{X_{i-1}} \to \mathbf{X'}$, $F'' : \mathbf{X'} \to \mathbf{X_i}$, where $\mathbf{X'}$ is the set of all possible configurations of layer σ'. We refer to such an operation as *decomposition* of the global map of layer i.

As is easy to see decomposition can always be applied to any global map F_i by trivial decomposition of corresponding local map, namely $f_i = \mathbf{1} \circ f_i$, where $\mathbf{1}$ is the identity mapping.

The reverse operation, *composition* of the global maps F_i and F_{i+1} is an exclusion of layer i without change of the operator transforming configurations $\mathbf{X_{i-1}}$ of layer $i-1$ to configurations $\mathbf{X_{i+1}}$ of layer $i+1$.

As well as with decomposition, composition can always be applied to any pair of global maps of particular FDCA by composition of their respective local maps.

Further in this paper we consider a particular case when these operations over global maps may be reduced to respective operations over their local maps. For this particular case we show that both composition and decomposition can reduce (and increase) complexity of respective FDCA.

3 Optimization of FDCA

In this section we utilize two IPPR problems to show that both operations can reduce (and increase) complexity of FDCA corresponding to the given problems. As we restrict our observation to the case when these operations can be reduced to the respective operations with corresponding local maps we consider now only decomposition and composition of local maps.

Firstly, we define the the local map computing the function with mesh-local properties (thus the local map computes some mesh-local function, but as we allow no knowledge about the structure of the function first, we compute this function by essentially global algorithms).

Consider the paving of 2D lattice $[n] \times [n] = N$ by squares $[n'] \times [n'] = N'$. We distinguish the subset x of the set of all configurations of a small square lattice and its complement $\overline{\mathbf{y}}$. We will distinguish some pattern of classes $\mathbf{y}$ and $\overline{\mathbf{y}}$ on the paving of N that is referred to as $\mathbf{Y}$, thus $|\mathbf{Y}| = |\mathbf{y}|^{\alpha|N|/|N'|} |\overline{\mathbf{y}}|^{(1-\alpha)|N|/|N'|}$ for some $0 < \alpha < 1$. Let f be the local map with the neighborhood N, mapping configurations $\mathbf{Y}$ into 1 and all the others into 0, $\Sigma_{i+1} = \{0, 1\}$ (Fig. 1). Recognition produced by f has inherently local properties defined by $\mathbf{y}$ and $\overline{\mathbf{y}}$ thus its computa-

tion by essentially global algorithm should cost

$$\mu(|N|, |\mathbf{Y}|), \tag{1}$$

where arguments of local map complexity μ are size of the configuration and the number of configurations that are to be recognized respectively (without going into details in this example we assume that second argument is the smallest of two inverse images of the respective local map). Therefore sequential and parallel time complexities of layer i, corresponding to the local map f are

$$|C|L_{i+1} = |C|\mu(|N|, |\mathbf{Y}|), \tag{2}$$
$$LT_{i+1} = \mu(|N|, |\mathbf{Y}|) \tag{3}$$

respectively. Using our knowledge about mesh-local structure of this IPPR problem we can decompose f into f' and f'' by addition of the new , intermediate layer. Let square N' be the basic neighborhood for the local map f' of the new layer, assuming $f' : \mathbf{y} \mapsto 1, \overline{\mathbf{y}} \mapsto 0$. For the local map f'' from the new layer i' to layer $i+1$ we use neighborhood $N \setminus B$, where B is the boundary that can be subtracted, so f'' maps class $\mathbf{Y}'$ of local configurations of the new layer i' that correspond to configurations $\mathbf{Y}$ of lattice N of layer i (that have to be recognized at layer $i+1$), to 1, and the others to 0 (Fig. 2). Local maps f' and f'' have complexities

$$\mu(|N'|, |\mathbf{y}|), \qquad \mu(|N \setminus B|, |\mathbf{Y}'|) \tag{4}$$

respectively. Thus we have the same operator between configurations of layer i and layer $i+1$, but the complexity of its realization now is

$$|C|(L_{i'} + L_{i+1}) = |C|(\mu(|N'|, |\mathbf{y}|) + \\ +\mu(|N \setminus B|, |\mathbf{Y}'|)), \tag{5}$$
$$LT_{i'} + LT_{i+1} = \mu(|N'|, |\mathbf{y}|) + \\ +\mu(|N \setminus B|, |\mathbf{Y}'|). \tag{6}$$

Assume the the boundary B has the size $o(|N|)$. The time of parallel computation (3) is shorter by factor $|C|$ compared with the time of sequential computation (2), while the time of sequential computation (5) of the decomposed global map can be shorter (we show it further) than the time

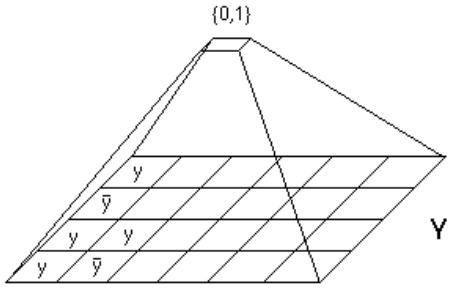

Figure 1: Local map before decomposition

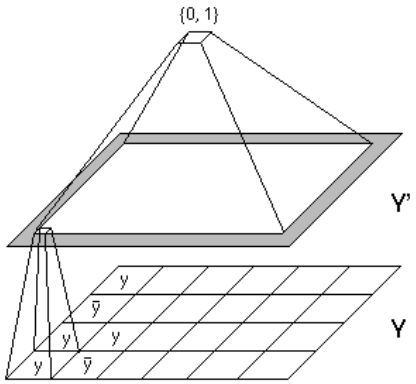

Figure 2: Result of decomposition

of sequential computation of the original global map (2) by factor $\exp(|C|)$. The source of the speed-up is that if class $\mathbf{Y}'$ is much smaller than $\mathbf{Y}$ then we can realize local maps using an algorithm (circuit) of complexity μ that depends mainly on the second argument, namely on the size of the class to be recognized ($\mathbf{Y}'$ in our case). Algorithms that are usually used in practical programming for realization of an arbitrary map, e. g. table look-up and binary search satisfy this condition. As for the schemes of circuit design for an arbitrary Boolean function, optimal size Lupanov's circuits don't satisfy the condition while disjunctive normal form-based (DNF-based) circuits satisfy [17], [18] (here we consider size of the circuit as its sequential time complexity μ).

From the other hand, decomposition cannot produce significant speed-up if local map is realized by the algorithm (circuit) of complexity

μ that depends mainly on the first argument, i.e. size of the local neighborhood. In this case $\mu(|N \setminus B|, |\mathbf{Y}'|) \approx \mu(|N|, |\mathbf{Y}|)$ and complexities of the decomposed global map (5), (6) cannot be much smaller than complexities of the original global map (2), (3) respectively.

Thus using the table look-up algorithm for realization of local maps we obtain for the original structure

$$LT_{i+1} = |N||\mathbf{y}|^{\alpha|N|/|N'|}|\overline{\mathbf{y}}|^{(1-\alpha)|N|/|N'|}, \quad (7)$$

and for the decomposed structure

$$LT_{i'} + LT_{i+1} = |N'||\mathbf{y}| + |N \setminus B||\mathbf{Y}'| \quad (8)$$

(similar expressions can be acquired for sequential time complexities). Assume that $|N'|$, $|\mathbf{y}|$, $|\overline{\mathbf{y}}|$, α are fixed, $N \to +\infty$, then

$$LT_{i+1} = |N|2^{\beta|N|} \quad (9)$$

for some positive β, while

$$LT_{i'} + LT_{i+1} \sim |N||\mathbf{Y}'|. \quad (10)$$

Under these constraints a speed-up produced by decomposition fully depends on the rate of growth of $\mathbf{Y}'$. We conjecture that for many practical IPPR tasks, the set of configurations $\mathbf{Y}'$ grows slowly, e. g. $|\mathbf{Y}'| = \text{poly}(|N|)$. Thus in this case we have polynomial versus exponential complexity of the FDCA after and before decomposition respectively.

Without detailed description we mention that similar estimates can be obtained for a decomposition speed-up considering size of Boolean circuit as complexity of local map. There usage of DNF-based circuit design of local map leads to the exponential decomposition speed-up (under similar constrains for parameters) even to compare with usage of asymptotically optimal size Lupanov's circuit for original local map.

We do not consider in this paper a possibility of non-trivial optimizing composition. As an example of a "good" application of composition operation we mention the trivial composition $f_i \circ \mathbf{1} \to f_i$. As is easy to see complexity of the resulting structure is less by $|C|\mu(\mathbf{1})$ for sequential time complexity and by $\mu(\mathbf{1})$ for parallel time complexity, where $\mu(\mathbf{1})$ is complexity of the identity mapping.

The investigated example shows that both decomposition and composition can lead to a reduction and, as they are inverse, to an increase of FDCA complexity.

4 Conclusion

The computational efficiency of the realization of mesh-local operators on mesh architectures caused not only by parallelism of computations but also by reduction of entropy (in our examples size of the class that is to be recognized) related to a local map. Thus entropy sensitive schemes for realization of local maps, e. g. table look-up, binary search, or DNF-based design of Boolean circuits, are effective in some cases. For a proposed framework, the use of a table look-up algorithm for realization of an arbitrary local map, shows the possibility of effective decomposition of local map in terms of the respective complexity measures.

The problem of optimal realization of operators by the proposed framework includes a number of new problems related with an approximation of operators by FDCA and a bounding complexity for particular classes of IPPR tasks.

Acknowledgments

This work was supported by the Research Grants Council of Hong Kong, project reference CityU 1039/96E.

The first author is greatly indebted to V. N. Kozlov for important discussions and to O. P. Skobtsov for his attention to the work.

References

[1] M. L. Brady, R. Raghavan and J. Slawny, "Probabilistic Cellular Automata in Pattern Recognition," *Proc. Inter. Joint Conf. Neural Networks*, Washington DC, USA, June 18–22, 1989, vol. 1, pp. 177–182.

[2] L. O. Chua and L. Yang, "Cellular Neural Networks," *Circuits and Systems*, vol. 2, pp. 985–988, 1988.

[3] I. Fajfar and F. Bratkovič, "Design of Monotonic Binary-Valued Cellular Neural Networks," *Proc. 1996 Fourth IEEE Inter. Workshop on Cellular Neural Networks and their Applications*, Seville, Spain, June 24–26, 1996, pp. 321–326.

[4] M. Garzon, *Models of massive parallelism. Analysis of cellular automata and neural networks.* Texts in Theoretical Computer Science, an EATCS Series. Berlin: Springer-Verlag, 1995.

[5] M. Garzon and S. Franklin, "Neural computability II (extended abstract)," *Proc. Inter. Joint Conf. Neural Networks*, Washington DC, USA, June 18–22, 1989, vol. 1, pp. 631–637, full version in *Journal of Artificial Neural Networks*, no. 1, 1994.

[6] G. Hernández and H. J. Herrmann, "Cellular Automata for Elementary Image Enhancement," *Graphical Models and Image Processing*, vol. 58, iss. 1, pp. 82–89, 1996.

[7] ISO/IEC JTC1/SC29/WG11, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/s, ISO/IEC CD11172-2," 1993.

[8] A. R. Khan, P. P. Choudhury, K. Dihidar, S. Mitra, P. Sarkar, "VLSI Architecture of a Cellular Automata Machine," *Computers Math. Applic.*, vol. 33, no. 5, pp. 79–94, 1997.

[9] K. R. Krieg, L. O. Chua and L. Yang, "Analog Signal Processing using Cellular Neural Networks," *Circuits and Systems*, vol. 2, pp. 958–961, 1990.

[10] O. Lafe, "Data Compression and Encryption Using Cellular Automata Transforms," *Engng. Applic. Artif. Intell.*, vol. 10, no. 6., 1998 (to appear).

[11] N. J. Leite and G. Bertrand, "A Parallel Image Processing Language Based on Computational Models," *Pattern Recognition*, vol. IV, pp. 181–184, 1992.

[12] C. Orovas and J. Austin, "Cellular Associative Neural Networks for Image Interpretation," *Proc. Sixth Inter. Conf. Image Processing and Its Applications*, Dublin, Ireland, July 14–17, 1997, vol. 2, pp. 665–669.

[13] A. Rosenfeld and A. C. Kak, *Digital Picture Processing.* New York: Academic, 1982.

[14] P. Sahota, M. F. Daemi and D. G. Elliman, "Training Genetically Evolving Cellular Automata for Image Processing," *Proc. Inter. Conf. Speech, Image Processing and Neural Networks*, Hong Kong, April 13–16, 1994, vol. 2, pp. 753–756.

[15] P. G. Tzionas, P. G. Tsalides and A. Thanailakis, "A New, Cellular Automaton-Based, Nearest Neighbor Pattern Classifier and Its VLSI Implementation", *IEEE Transactions on VLSI Systems*, vol. 2, no. 3, pp. 343–353, September 1994.

[16] L. Uhr (ed.), *Parallel Computer Vision.* Academic Press, 1987.

[17] I. Wegener, *The Complexity of Boolean Functions.* John Wiley & Sons, 1987.

[18] С. В. Яблонский, *Введение в дискретную математику.* Москва: Наука, 1986. English translation: S. V. Yablonsky, *Introduction to discrete mathematics. Translated from the Russian by Oleg Efimov.* Moscow: Mir, 1989.

[19] L. Yang, L. O. Chua, K. R. Kreig, "VLSI Implementation of Cellular Neural Networks", *Circuits and Systems*, vol. 3, pp. 2425–2427, 1990.