EXTREMAL AUTOMATA FOR IMAGE SHARPENING

GONZALO HERNANDEZ*, HANS J. HERRMANN
HLRZ, Forschungszentrum Jülich
Postfach 1913, D-5170 Jülich, Germany

and

ERIC GOLES
Departamento de Ingeniería Matemática, U. de Chile
Casilla 170-3, Santiago, Chile

-Received 6 June 1993
Revised 12 August 1993

We study numerically the parallel iteration of Extremal Rules. For four Extremal Rules, conceived for sharpening algorithms for image processing, we measured, on the square lattice with Von Neumann neighborhood and free boundary conditions, the typical transient length, the loss of information and the damage spreading response considering random and smoothing random damage. The same qualitative behavior was found for all the rules, with no noticeable finite size effect. They have a fast logarithmic convergence towards the fixed points of the parallel update. The linear damage spreading response has no discontinuity at zero damage, for both kinds of damage. Three of these rules produce similar effects. We propose these rules as sharpening algorithms for image processing.

Keywords: Extremal Automata; Image Processing; Sharpening.

1. Introduction

In this work we study the parallel iteration of Extremal Rules (ER). These rules were defined in Ref. 1 as a generalization of FES Rules (Forced-Move and Stay constraints), and they were introduced in Refs. 3 and 4 as image processing algorithms.

Let $G = (V, E)$ be an undirected, connected, finite graph ($V$ set of vertex and $E$ set of edges), with $|V| = N$. To each vertex or site $i \in V$ we assign values $x_i \in Q = \{0, \ldots, q - 1\}$, with $q \geq 3$ (think of colors in an image). We will say that $f_i$, the local transition function on vertex $i$, is an Extremal Rule if:

$$f_i(x_j : j \in V_i) \in \{m_i, x_i, M_i\} \quad \text{(1)}$$

*Permanent address: Departamento de Ingeniería Matemática, U. de Chile.
E-mail: gherman@dim.uchile.cl
and
\[ x_i \notin ]m_i, M_i[ \Rightarrow f_i(x_j : j \in \mathcal{V}_i) = x_i, \]  
(2)
where \( \mathcal{V}_i \) is the neighborhood of vertex \( i \), i.e.,
\[ \mathcal{V}_i = \{ j \in V : (i, j) \in E \text{ and } j \neq i \} \]  
(3)
and
\[ m_i = \min_j \{ x_j : j \in \mathcal{V}_i \}, \quad M_i = \max_j \{ x_j : j \in \mathcal{V}_i \} \]  
(4)
are the local extreme values.
This means that a node does not change when it is outside the range defined for the maximum and minimum states represented in his neighborhood. Otherwise it takes the maximum or the minimum value.

In this paper we study numerically the parallel iteration of ER. For this purpose we take the following ER: Nearest Extremum Rule (NE)\(^2\)\(^-\)\(^4\) and Potts (PR), Mean (MR) and Less Number of States (LN), that were introduced in Ref. 1. We measure, for the parallel iteration, the typical transient length, the loss of information and the damage spreading due to random and smoothening damage. Finally, we compare the PR, MR and LN rules with the NE rule in order to use these rules as image processing algorithms for sharpening images.

2. The Extremal Rules

For the numerical simulations, we consider the following rules:

**Nearest Extremum Rule: NE**

\[ f_i(x_j : j \in \mathcal{V}_i) = \begin{cases} m_i & \text{if } x_i - m_i < M_i - x_i \text{ and } x_i \in ]m_i, M_i[ \\ M_i & \text{if } M_i - x_i > x_i - m_i \text{ and } x_i \in ]m_i, M_i[ \\ x_i & \text{otherwise.} \end{cases} \]  
(5)

In this rule, the sites must take the value of the nearest extreme.

**Potts Extremum Rule: PR**

\[ f_i(x_j : j \in \mathcal{V}_i) = \begin{cases} m_i & \text{if } C(m_i) > C(M_i) \text{ and } x_i \in ]m_i, M_i[ \\ M_i & \text{if } C(M_i) > C(m_i) \text{ and } x_i \in ]m_i, M_i[ \\ x_i & \text{otherwise,} \end{cases} \]  
(6)

where
\[ C(m_i) = |\{ j \in \mathcal{V}_i / x_j = m_i \}| \quad \text{and} \quad C(M_i) = |\{ j \in \mathcal{V}_i / x_j = M_i \}|. \]  
(7)

In this case, the evolution goes locally to the more represented extreme.

**Mean Extremum Rule: MR**

\[ f_i(x_j : j \in \mathcal{V}_i) = \begin{cases} m_i & \text{if } S_i > 0 \text{ and } x_i \in ]m_i, M_i[ \\ M_i & \text{if } S_i < 0 \text{ and } x_i \in ]m_i, M_i[ \\ x_i & \text{otherwise,} \end{cases} \]  
(8)
where

$$S_i = \sum_{j \in \mathcal{V}_i} x_j - |\mathcal{V}_i| x_i. \quad (9)$$

In this rule, a site goes to the minimum value of its neighborhood if its value is smaller than the mean value of its neighborhood. Otherwise, it takes the maximum.

**Less Number of States Rule: LN**

$$f_i(x_j : j \in \mathcal{V}_i) = \begin{cases} M_i & \text{if } |\{ j \in \mathcal{V}_i / x_j \geq x_i \}| < |\{ j \in \mathcal{V}_i / x_j < x_i \}| \\ m_i & \text{if } |\{ j \in \mathcal{V}_i / x_j \geq x_i \}| > |\{ j \in \mathcal{V}_i / x_j < x_i \}| \\ x_i & \text{otherwise.} \end{cases} \quad (10)$$

In this rule a site takes the extreme of the most represented interval between: \([m_i, x_i]\) and \([x_i, M_i]\).

This rule is equivalent to the following one:

$$f_i(x_j : j \in \mathcal{V}_i) = \begin{cases} M_i & \text{if } 2|\{ j \in \mathcal{V}_i / x_j \geq x_i \}| < |\mathcal{V}_i| \\ m_i & \text{if } 2|\{ j \in \mathcal{V}_i / x_j \leq x_i \}| < |\mathcal{V}_i| \\ x_i & \text{in tie cases.} \end{cases} \quad (11)$$

Because

$$|\{ j \in \mathcal{V}_i / x_j \geq x_i \}| + |\{ j \in \mathcal{V}_i / x_j < x_i \}| = |\mathcal{V}_i| \quad (12)$$

These rules satisfy Eqs. (1) and (2), and thus, they are ER.

3. **Results**

Our objective is to measure the performance of the local rules, defined in Sec. 2, as sharpening algorithms in image processing. For this purpose, we used these rules to enhance contrast in a digitized 2-d image represented on a 2-d square lattice with: Von Neumann's neighborhood (nearest neighbors), free boundary conditions and \(N = n^2\) sites or pixels, which can take every state or gray scale colors in \(Q = \{0, \ldots, q - 1\}\), \(q \geq 3\). We applied the local rules and studied the typical transient length, the loss of information and the damage spreading, comparing the rules. In the following, we define these quantities and present the results.

We used the Connection Machine CM2 at the GMD in Bonn to make the computations with linear lattice sizes between 128 and 1024, a set of 5,000 to 10,000 random uniform initial configurations and \(q = 256\) states. We get a performance of 25 mega-updates per second, which is not very high due to the maximum and minimum local computations needed for the rules.

3.1. **Typical transient length**

This quantity is the number of updates which takes the algorithm to reach the steady state. Then, we can determine which is the fastest algorithm and which is
the convergence speed. For that, we averaged the transient length over a set of random initial configurations.

We observe a very fast convergence rate to fixed points for all the rules. For the average transient length $\ell$, we find the following convergence speed:

$$\ell = a \cdot \log(n) + b$$

(13)

where the constants $a$, $b$ depend on the rule and $n$ is the linear size of the lattice.

The fastest algorithm is LN and the slowest one is PR, with a marked difference. The NE and MR algorithms have almost the same rate of convergence. We exhibit this behavior in Fig. 1.

![Graph showing average length of transient as a function of system size for four different rules.](image)

Fig. 1. Average length of transient as a function of system size for the four different rules.

### 3.2. Loss of information and qualitative effects

We measured the loss of information and the qualitative effects of the rules by two quantities: The mean square deviation of the frequency of states in the final configuration from the initial one, $\Delta$, which we take uniformly randomly distributed, and the mean Hamming distance between the initial and final configuration, $\Delta'$. That is to say, we measured, for all the rules, the following quantities over a wide set of random initial configurations (random images):

$$\Delta = \frac{1}{q \cdot |C|} \sum_{c} \sum_{i=0}^{q-1} (\lambda_i - 1/q)^2$$

(14)

$$\Delta' = \frac{1}{N \cdot |C|} \sum_{c} \sum_{i=1}^{N} \{1 - \delta[C_i(t = 0), C_i(t = \infty)]\}$$

(15)
where \( C \) denotes the set of steady configurations obtained from a random initial one, \( \lambda_i \) is the frequency of state \( i \) in the final configuration and finally, \( (C_i(t = 0)) \) and \( (C_i(t = \infty)) \) are the initial and final configurations.

It is clear that the first quantity, \( \Delta \), correspond to the relative deviations between the frequencies of states and it is a measure of the qualitative effect of the rules. The second quantity, \( \Delta' \), measures how different the end configuration is from the initial one.

In Fig. 2 we show the normalized \( \Delta \) defined in Eq. (14). This quantity goes to a constant, that depends on the rule, as the lattice size is increased. The MR and PR algorithms cause greater effect on images than the LN and NE rules, in that sense there is a larger difference in the frequency of the states. The LN rule does not cause a great change on the image. For all the rules the \( \Delta' \) quantity takes values between 0.71 and 0.78, i.e., 71 to 78 percent of the sites change their values under the action of the rules. As a function of the lattice size \( \Delta' \) shows very small variations, which means no noticeable finite size effects. The \( \Delta' \) values are shown in the following table:

\[
0.771 \leq \text{NE Rule} \leq 0.776 \quad 0.722 \leq \text{PR Rule} \leq 0.725 \\
0.738 \leq \text{MR Rule} \leq 0.743 \quad 0.716 \leq \text{LN Rule} \leq 0.719
\]

The NE Rule causes the greater effects considering both quantities on the random images.

![Graph showing \( \Delta \) as a function of system size for four rules](image)

Fig. 2. Mean square frequency deviation \( \Delta \) as a function of system size for the four different rules.

### 3.3. Damage spreading

Another interesting feature of an algorithm is its stability. Inquiring how different is the result if we make a small perturbation on the initial configuration, and how
stable are the convergent configurations with respect to small perturbation gives information about stability.

For our rules, we study damage spreading by making two types of initial perturbations: A randomizing one and a smoothening one. In the first case, we take a random initial configuration, choose at random a fixed fraction \( p \) of sites and change its states to another one. Then, we have two initial configurations: One random and one randomly damaged. We apply the same rule in both cases and measure the Hamming distance between the convergent configurations in each case, defined by:

\[
H_d(C^1, C^2) = \frac{1}{N} \sum_{i=1}^{N} [1 - \delta(C^1_i, C^2_i)],
\]

where

\[
\delta(u, v) = 1 \text{ if } u = v \text{ and } 0 \text{ otherwise.}
\]

This distance describes the fraction \( p \) of sites in the final configurations that are different, and this is a measure of the stability of the algorithm. This quantity is called the final damage.

In the second case, the random smoothening damage, we do the same thing, except that the damage in the initial configuration is different. We choose sites at random and replace their state by the neighborhood’s average. This damage measures a perturbation acting to the opposite effect of the extremal rules: A smoothening procedure.

We perform the random and smoothening damage spreading study on a lattice of linear size equal to 512. For other sizes the results are similar.

![Fig. 3. Spreading of random damage for different rules.](image-url)
Figures 3 and 4 show the behavior of the rules when different fractions $p$ of random and smoothening damage are applied. The "Percent of Final Damage", $100 \cdot H_d$ at $t = \infty$, is the percent of different sites between final configurations in the damaged and not damaged cases.

A linear response without any discontinuity at the origin was found for both cases. In the randomly damaged case, the slopes are greater than in the smoothening case. This means more stability of the rules in comparison to a smoothening damage.

![Graph showing percent of final damage versus $p$](image1)

**Fig. 4.** Spreading of smoothening damage for different rules.

![Graph showing fraction of different sites versus $n$](image2)

**Fig. 5.** Comparison between the Extremal Rules: Hamming distance between two pairwise chosen final configuration starting at the same initial configuration.
3.4. Comparison

Finally, we compare the four rules by starting with the same initial configuration and by measuring pairwise the Hamming distance between the final configurations of all the other ones. We choose this criterion because we want to measure the differences between the rules.

The PR rule causes a very different effect than the other rules (above 70 percent of difference). The closest rules are MR and LN, with no less than 10 percent of different sites. We also observe no finite size effects. In Fig. 5 we show these results.

4. Conclusions

In this work we study numerically the parallel update of four extremal rules. By using the CM2, we simulate these rules on a square lattice with a Von Neumann neighborhood and free boundary conditions.

We found:

1. Very fast convergence to fixed points at a logarithmic rate in the linear size of the lattice. The fastest algorithm is LN and the slowest one is PR. The rates of convergence of NE and MR are similar, but MR is faster.

2. Linear response to a random and a smoothening damage, without any discontinuity at zero damage. The stability of each algorithm depends on which kind of damage is applied. Moreover, by considering the two kinds of damage, the more stable rule is MR.

3. Three of the rules, NE, MR and LN produce similar effects on the random uniform configurations. The PR rule produces a totally different effect. The more similar rules are LN and MR, and the most different are NE and PR. We can observe no significant finite size effect. All the rules produce great effects on random images that we measured by the changes on the frequency of states (gray scale colors), and Hamming distance between initial and final configuration. The greater effect is produced by the NE and MR rules, the LN algorithm produces smaller changes.

From previous remarks we conclude that MR and NE rules are good sharpening algorithms for image processing on random images. The MR rule is also faster and more stable under random and smoothening perturbations than the NE rule.

We are presently investigating the effect of the rules presented in this paper on real two dimensional images.

Acknowledgments

This work was partially supported by PG-04192 U. de Chile, C-10003 Fundación Andes and EEC. G. H. wishes to acknowledge financial support of the EEC through a "Marie Curie" fellowship.
References


