

## CA-GIS 통합 시스템을 이용한 GIS 연산의 구현

박수홍\*

### A prototype implementation of GIS operations using an integrated CA-GIS and evaluation of their relative performance

Soo-hong Park

#### 요 약

본 연구는 CA 시뮬레이터가 GIS의 분석 기관으로 결합되어 있는 통합 CA-GIS를 이용하여 기존의 GIS 연산들을 구현하고 연산들의 성능을 향상시키며, 현재의 GIS에서는 구현이 불가능하거나 어려웠던 연산들을 새로이 개발하는데 그 목적을 두었다. 대다수의 국지 및 근린 GIS 연산들이 통합 CA-GIS에 의해 성공적으로 구현될 수 있었으며, 이 중 반복적인 프로세싱을 요구하는 일부 연산들은 CA가 갖고 있는 우수성에 기인해 원래 연산보다 더 나은 성능을 보여주었다. 또한 새로운 유형의 연산들 (다양한 형태의 공간 필터, 시계열 필터, 확산 연산자 등)이 시험 제작되었고 다차원 공간연산, 시공연산 그리고 동적 연산들이 이 통합 시스템을 사용하여 구현될 수 있음이 예시되었다.

**ABSTRACT :** This paper demonstrates the prototype implementation of existing GIS operations using an integrated CA-GIS system where a CA simulator serve an analytical engine for GIS and compares the performance of the operations implemented here with the original ones. In addition, some new GIS operations that are not available in contemporary GIS are implemented and tested within the integrated system. It has shown that most of the local and focal operations were successfully implemented with CA-GIS and some operations written with the CA simulator showed better performance. New types of operations such as various shape filters, time-series filter, and diffusion operators were able to be implemented using the CA-GIS, which appears to be significant to enhance deficient analytical capabilities of contemporary GIS.

#### 서 론

셀룰라 오토마타(Cellular Automata, CA)는 복잡한 동적 시스템의 문제를 해결하는데 사용되는 일련의 방법론이다. 이 셀룰라 오토마타는 여

러 과학 분야에서 널리 연구되어 왔으며, 다양한 동적 모델의 프로세서를 모의 실험하는데 유용한 도구로 사용되어 왔다(White and Engelen 1994). CA는 특성상 래스터 GIS와 유사하고 현재 GIS가 안고 있는 여러 문제들을 해결하는데 많은 잠재력

\* 책임연구원, 미국 인디애나 대학 CIPEC(Center for the Study of Institutions, Population, and Environmental Change), 408 North Indiana Avenue, Bloomington, Indiana 47408, USA (812) 855-7134.

을 갖고 있는 것으로 평가되어 두 시스템의 통합에 대한 논의가 활발히 이루어져 왔다 (Itami 1988; Itami and Clark 1992; White and Engelen 1993, 1994; Wagner 1994, 1997). 최근에 이러한 통합 시스템의 모형이 개발되어(Park 1996) GIS 연산과 동적 공간 모델(dynamic spatial models)들을 이 통합 시스템을 사용하여 손쉽게 구현할 수 있게 되었다. 본 연구에서는 통합 시스템을 이용하여 기존의 래스터 GIS 연산들을 구현하고, 새롭게 구현된 이 연산들의 성능(performance)을 원래의 GIS 연산들과 비교하며, 더 나아가 현재 GIS에서는 구현되기 어려운 새로운 GIS 연산들을 시험적으로 구현하는데 그 목적을 두었다.

## 셀룰라 오토마타(Cellular Automata)

셀룰라 오토마타는 1950년대 초 John von Neuman과 그의 동료 Stanislaw Ulam에 의해 개발되어(von Neuman 1966), 물리학, 컴퓨터 공학, 생물학, 수학 등을 포함한 여러 자연과학 또는 공학분야에서 다양한 동적 현상들을 연구하는데 사용되어 왔다. CA의 원리 및 일반적인 특성은 무질서 시스템(chaotic systems)들을 연구하는데 있어 중요한 주제로 간주되고 있으며, 동적이고 복잡한 자연현상들을 모의 실험하는데 유용한 도구로 사용되어 왔다

셀룰라 오토마타는 이산적인 시공(discrete time and space)에 기초한 비선형 동적 모델로 정의될 수 있다(Wolfram 1984a; Gutowitz 1991). CA 모델에서 기본 단위는 하나의 셀(cellular automaton)이며, 셀의 상태는 매시간 모든 셀에 공통적으로 적용되는 변화규칙(transition rule)에 따라 설정된다. 즉, 각 셀의 상태(즉, 지역상태)는 일정한 거리나 방향에 의해 구성되는 주변셀(neighborhood)들의 상태에 의해 변화하는데, 이러한 주변

셀들의 물리적 구성과 변화방법은 변화규칙에 의해 명확히 정의된다. 매시간 개개의 셀 상태(값)는 공통적이고 지역적인 변화규칙에 의해 새로운 상태로 변화하게 되고, 이 상태는 다시 그 다음번 단계의 초기상태로 입력되어 반복적으로 연산이 진행된다.

## CA의 구성요소와 래스터 GIS와의 유사성

CA 모델은 크게 네 가지 기본 요소로 구성된다. CA의 첫 번째 요소는 셀룰라 공간 (cellular space)으로 그 분할방법(tessellation)에는 제한이 없다. 따라서 어떠한 정규 다각형으로도 분할이 가능하다. 보통 삼각형, 사각형 혹은 육각형이 많이 사용되는데, 이 중에서도 사각형은 컴퓨터에서 구현하기가 쉬워 널리 사용되고 있다. 육각형은 가스확산이나 유체흐름(turbulent flow of liquids)에 많이 사용되어 왔다(Wilson 1988). 이론적으로 셀룰라 공간은 다차원으로 구성될 수 있으나, 많은 연구들이 주로 1차원(Wolfram 1984b)이나 2차원 CA 모델에 관해 이루어져 왔다. 2차원 셀룰라 공간은 래스터 GIS의 레이어에 해당한다.

CA의 두 번째 구성요소인 지역상태(local states)는 주어진 시간 각 셀의 상태를 나타내는데, 보통 숫자로 표현된다. CA의 대표적인 예로 널리 알려진 생명게임(game of life) (Gardner 1971)에서 각 셀은 0 (사망) 혹은 1 (생존)의 값으로 그 상태가 표현된다. 래스터 GIS에서 각 래스터의 값과 CA의 지역상태는 개념적으로 일치한다. 그러나 래스터 GIS가 바이트(byte), 정수(integer) 및 실수(real or floating point number)를 모두 다룰 수 있는 반면에 CA가 이산구조에 기초하는 관계로 CA의 데이터형은 바이트나 정수형에 국한된다.

세 번째 요소는 주변셀(neighborhood)이다. 주변셀은 중심셀(focus cell) 주위에 인접하고 있는

셀들의 집합을 말하는데, 보통 거리, 방향, 혹은 각도에 의해 결정된다. 흔히 사용되는 주변셀들의 구성은 Figure 1과 같다. 이는 영상처리(image processing)나 GIS에 사용되는 공간필터(spatial filter)의 마스크(mask)나 윈도우(roving window)와 개념적으로 매우 유사하다. CA에서 주변셀의 구성에 제한이 없기 때문에 다양한 형태나 크기의 구성이 가능하다. 또한, 중심셀이 주변셀의 집합에 포함되기도 하고 주변셀의 구성이 시간차원(temporal)으로도 확대될 수 있다.

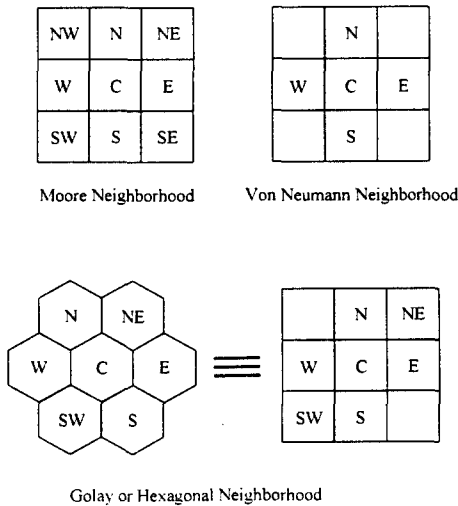


Fig. 1 Neighborhoods for two-dimensional CA(After Allison and Sales 1992)

마지막 CA의 구성요소는 변화규칙(transition rule)이다. 변화규칙은 개개의 셀이 매 시간별로 어떻게 변화할 지를 규정하는데, 여기에 주변셀들의 구성과 위치가 정의되어 있다. 이러한 변화규칙은 보통 특정한 컴퓨터 언어(예를 들면 Forth)나 셀룰라 언어에 의해 기술된다. 셀룰라 언어는 CA 시뮬레이터와 함께 제공되어 CA 모델들을 기술하고 실행하는데 사용된다. 본 연구에서 사용된 CA 시뮬레이터, Cellular(Eckart 1995)는 Cellang이라는 셀

룰라 언어와 그 컴파일러를 제공하고 있다.

## CA의 연구동향

지리학계에서는 Tobler(1979)에 의해 처음으로 CA가 소개 되었으나, 그 이후 소수의 연구자들에 의해서만 논의되어 왔다(Couclelis 1985, 1987, 1988)., 최근에 들어와 CA 기법을 이용한 다양한 공간모델들이 발표되면서 그 활용예와 가능성이 논의되어 왔다. 지금까지 발표된 응용 사례로서는 산불모델링(Green 1982; Clarke et al. 1994), 토지이용 변화 모델링(White and Engelen 1993, 1994), 기후변화 모델링 (White and Engelen 1994), 촌락 주거지 변화 (Deadman et al. 1994), 도시성장 모델링 (Batty et al. 1989; Batty and Xie 1994; Kirtland et al. 1994), 식생변화 모델링 (Green et al. 1985; Silvertown et al. 1992) 등이 있다.

이와 아울러 현재 GIS가 갖고 있는 동적 모델링 기능을 향상하기 위해 일부 연구자들은 CA-GIS 통합시스템의 구현 가능성을 논의해왔다 (Itami 1988; Itami and Clark 1992; White and Engeln 1994; Wagener 1994, 1997). 그들은 이 통합시스템이 현재 GIS에 미약한 동적 모델링 기능을 추가하고 연산의 성능을 향상하며 보다 효율적인 시공연산(spatio-temporal operations)의 개발에 기여할 것이라고 주장해왔다. 그러나 최근까지도 성공적인 통합시스템 개발되지 않았으나 필자의 선행연구(Park 1996)에 의해 CA-GIS 통합시스템의 모형이 개발되어 그 가능성이 입증되었다.

## CA-GIS 통합 시스템의 구조

CA-GIS 통합 시스템은 CA 모델들을 기술하고 실행하는데 필요한 CA 시뮬레이터와 래스터 GIS

가 통합 유저 인터페이스와 연결모듈들, 즉 자료 변환기(data model translator)와 연산처리기(operation handler)에 의해 결합되어 있는 구조를 지니고 있다 (Figure 2). 이 시스템은 PC환경에서 개발되었는데, 래스터 GIS로는 IDRISI (Eastman 1992)가 선택되었고 CA 시뮬레이터로는 Cellular (Eckart 1995)가 사용되었다. IDRISI와 Cellular는 모두 모듈화된 구조로 되어 있으며, 특히 Cellular는 CA 모델들을 프로그램하고 컴파일 할 수 있도록 Cellang이라는 셀룰라 언어를 제공하고 있어 이를 통해 공간모델이나 GIS 연산들을 구현할 수 있다. Cellang은 Figure 3에서 보는 바와 같이 C 또는 PASCAL 프로그래밍 언어와 아주 유사한 문법구조를 갖고 있다.

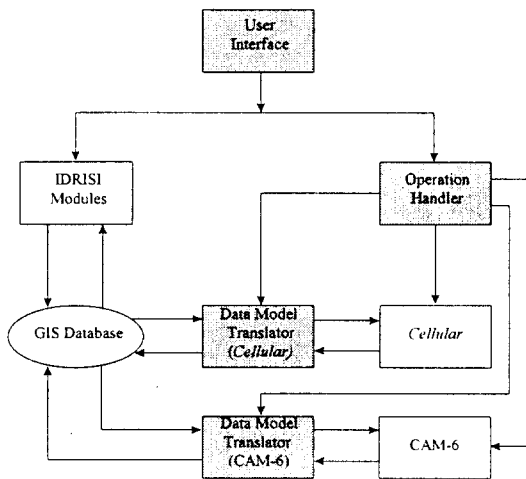


Fig. 2 Structure of the integrated system

```

2 dimensions of 0..1
sum := [0, 1] + [1, 1] + [1, 0] + [-1, 1] + [-1, 0] +
        [-1, -1] + [0, -1] + [1, -1]
cell := 1 when (sum = 2 & cell = 1) | sum = 3
        := 0 otherwise
    
```

Fig. 3 The Game of Life written in Cellang (Eckart 1995)

통합 유저 인터페이스는 기존의 IDRISI 유저 인터페이스와 유사한 구조를 지니고 있으나 새로운 모듈의 추가 및 삭제가 자유롭고 기존의 IDRISI의 메뉴를 재구성할 수 있는 장점이 있다. 사용자는 이 통합 유저 인터페이스를 통해 IDRISI나 CA 시뮬레이터로 작성된 모델 또는 GIS 연산들을 수행할 수 있다. 두 연결모듈(자료구조 변환기와 연산 처리기)은 GIS 데이터베이스에서 자료를 전송하여 이를 CA 시뮬레이터에서 사용할 수 있도록 변환해주고 CA 시뮬레이터를 실행하는 기능을 갖고 있다. 즉, GIS 데이터베이스에 저장된 래스터 레이어들은 CA 시뮬레이터에서 사용되기 위해 자료변환기에 의해 그 구조가 변환기에 의해 번역되고, 연산처리기는 CA 시뮬레이터를 실행하는데 필요한 입력자료들을 사용자로부터 입력받아 실제로 연산을 수행한다. CA에서 처리가 끝나면 그 결과는 화면에 출력되거나 다시 데이터베이스의 새로운 레이어로 저장되어 그 이후의 GIS 분석에 사용할 수 있게 된다 (Figure 4).

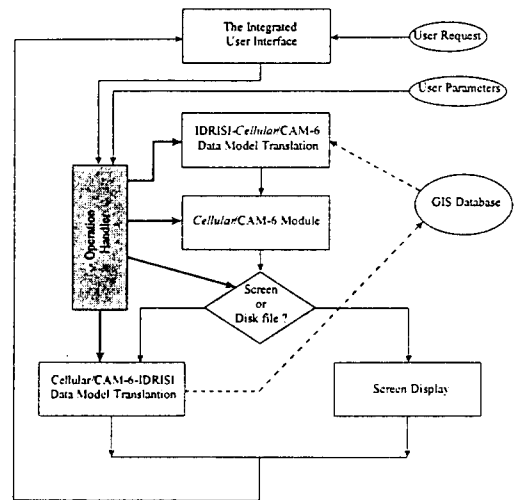


Fig. 4 Architecture of the operation handler

## 통합 시스템을 이용한 기존 GIS 연산의 구현

래스터 GIS의 기능을 분류하고 정의하는데는 여러 가지 방법이 있지만, 이 중에서도 가장 널리 사용되는 접근방법으로는 지도 모델링(cartographic modeling) (Tomlin 1990)을 들 수 있다. 이 방법론은 지도형태로 표현된 공간 데이터의 분석과 모델링 기법들을 제시하고 있다. 지도 모델링은 엄격히 말해서 래스터 GIS가 아니지만 래스터 GIS의 연산들을 정의하고 있고, 대부분의 래스터 GIS가 지도 모델링 방법론에서 소개된 연산들을 구현하고 있다. 따라서 본 연구에서는 Tomlin의 지도 모델링 방법론을 현재 GIS의 연산들을 분류하고 통합 시스템을 사용하여 새로운 GIS 연산들을 설계하는데 개념적 틀로 사용되었다.

Tomlin(1990)에 따르면, 래스터 GIS 연산들은 크게 네가지 연산 그룹으로 나눌 수 있다. 이 네가지 연산 그룹은 다음과 같다.

- 국지 연산(local operations)
- 근린 연산(focal operations)
- 증가 연산(incremental operations)
- 구역 연산(zonal operations)

국지 연산(local operations)은 단일 혹은 복수 레이어에 있는 개개의 셀에 부여되는 연산으로, 각 셀의 값을 처리하여 새로운 레이어의 동일한 위치에 그 값을 출력한다. 국지연산은 사용되는 레이어의 수에 따라 두 유형 (단일 레이어 국지 연산과 복수 레이어 국지 연산)으로 나뉜다. 단일 레이어에 적용되는 국지 연산에는 크게 재분류(local reclassification), 수학적함수, 그리고 상수함수 (scalar functions) 등이 있다. Table 1은 CA-GIS 통합 시스템의 단일 레이어 국지 연산들을 나타내고 있는데, 첫 번째 행은 IDRISI에 의해 제공되는 국지 연산이고, 두 번째 행은 CA 본 연

구에서 CA 시뮬레이터에 의해 구현된 단일 레이어 국지 연산이며, 마지막 행은 CA-GIS 통합 시스템이 결과적으로 갖게 되는 단일 레이어 국지 연산들의 리스트이다.

Table 1. Local operations on a single layer

Local operations defined by Tomlin (1990)	Local operations in IDRISI	Local operations implemented in the CA simulators	Local operations in the integrated CA-GIS system
LocalRating (Reclassification)	✓		✓
LocalMaximum (Greater Than)		✓	✓
LocalMinimum (Less Than)		✓	✓
LocalSum (Scalar Add)	✓	✓	✓
LocalDifference (Scalar Subtract)	✓	✓	✓
LocalProduct (Scalar Multiply)	✓	✓	✓
LocalRatio (Scalar Divide)	✓	✓	✓
LocalRoot	✓		✓
LocalSine	✓		✓
LocalCosine	✓		✓
LocalTangent	✓		✓
LocalArcSine	✓		✓
LocalArcCosine	✓		✓
LocalArcTangent	✓		✓

재분류 연산(local rating or reclassification)은 보통 사용자로부터 특정한 범위를 입력받아 각 셀의 값을 분류하게 되는데, 이 과정은 CA 시뮬레이터에 의해 구현되기 쉽지않고 수학적함수들(삼각함수, 지수함수, 로그, 제곱근 등)이 CA 시뮬레이터에서 지원되지 않기 때문에, 이들은 제외되었다. Figure 5는 단일 레이어 국지 연산 중의 하나인 논리적 대소 연산자의 Cellang코드이다.

```

2 dimensions of
  inlayer, scaleval of -32768..32767
end

if cell.inlayer > cell.scaleval
  then cell.inlayer := cell.inlayer
  else cell.inlayer := cell.scaleval
end
    
```

Fig. 5 Cellang code for local maximum (greater than) operation

국지 연산은 복수 레이어에도 적용될 수 있다. 복수 레이어에서 가능한 국지연산으로는 재분류 (reclassification), 교차분류 (cross-reclassification), 통계연산 중첩연산(overlay) 등이 있다. 재분류 및 교차 분류 연산은 앞서 설명된 바와 같이 CA 시뮬레이터에서 사용자에게 의해 정의되는 범위를 입력하는데 어려움이 있고 모든 연산이 CA에서는 국지적으로 이루어지는 관계로 전역변수(global variable)를 사용할 수 없어 구현되지 못하였다. 통계 연산에서는 평균, 최소, 최대값 연산등이 구현되었다. 마지막으로 중첩 연산에서는 네 가지의 기본 연산(+, -, \*, /)과 커버(cover)연산이 구현되었다. Figure 6과 7은 각각 국지 평균 연산과 커버 연산의 실제 Cellang 코드이다. Table 2는 본 연구에서 개발된 복수 레이어에 적용될 수 있는 국지 연산들의 리스트이다.

근린 연산(focal operations)과 증가 연산 (incremental operations)은 각 셀의 값을 주변 (neighborhood)에 있는 셀들에 의해 결정하는 연산들로, CA 모델과 개념적으로 일치한다. 증가 연산들은 여기에서 근린 연산의 특별한 경우로 간주되었다. 근린 연산은 주변셀의 유형에 따라 크게 두 그룹으로 나뉘는데, 그 첫 번째 그룹은 공간적으로 인접한 주변셀에 바탕을 둔 근린 연산이다. 증가 연산자는 공간적으로 인접한 주변셀들을 사용하고 새로운 셀의 값을 주변셀들의 기하학적 특성에 의해 결정하기 때문에 이 그룹에 속한다. 통합 시스템 내에서 모두 14개의 근린 연산자가 구현될 수 있었는데 (Table 3), 여기에는 근린 최대치(focal maximum), 근린 비율(focal percentage), 근린 백분율(focal percentile), 근린 합(focal sum) 그리고 근린곱(focal product)등이 포함되어 있다. Figure 8은 경사도(incremental gradient)연산의 실제 Cellang 코드이다.

두 번째 그룹의 근린 연산은 보다 확장된 주위 셀들을 사용하는 연산들로 주변셀들이 중심셀로

```

dimensions of
  value[] for 5 of -32768..32767
  outlayer of -32768..32767
end

sum := 0

forall i:0..4
  sum := sum + cell.value[i]
end

cell.outlayer := sum / 5
    
```

Fig. 6 Cellang code for local mean operation

```

2 dimensions of
  11,12 of -32768..32767
  outlayer of -32768..32767
end

cell.outlayer := cell.12 when cell.11 = 0
               := cell.11 otherwise
    
```

Fig. 7 Cellang code for cover operation

Table 2. Local operations on multiple layers

Local operations defined by Tomlin (1990)	Local operations in IDRISI	Local operations implemented in the CA simulators	Local operations in the integrated CA-GIS system
LocalRating (Cross Reclassification)			
LocalCombination (Cross Combination)	✓		✓
LocalVariety			
LocalMajority		✓	✓
LocalMinority		✓	✓
LocalMaximum	✓	✓	✓
LocalMinimum	✓	✓	✓
LocalSum (Overlay Add)	✓	✓	✓
LocalDifference (Overlay Subtract)	✓	✓	✓
LocalProduct (Overlay Multiply)	✓	✓	✓
LocalRatio (Overlay Divide)	✓	✓	✓
Cover	✓	✓	✓
Local Mean		✓	✓

Table 3. Focal operations of immediate neighborhoods

Focal operations defined by Tomlin (1990)	Focal operations in IDRISI	Focal operations implemented in the CA simulators	Focal operations in the integrated CA-GIS system
FocalRating			
FocalCombination			
FocalVariety			
FocalMajority	✓	✓	✓
FocalMinority		✓	✓
FocalMaximum		✓	✓
FocalMinimum		✓	✓
FocalSum		✓	✓
FocalProduct		✓	✓
FocalMean	✓	✓	✓
FocalPercentage		✓	✓
FocalPercentile		✓	✓
FocalRange		✓	✓
FocalMeanDeviation		✓	✓
FocalRanking		✓	✓
FocalInstability (Group)	✓		✓
IncrementalLinkage			
IncrementalLength			
IncrementalPartition			
IncrementalFrontage (Perimeter)	✓		✓
IncrementalArea (Area)	✓		✓
IncrementalVolume			
IncrementalGradient (Slope)	✓	✓	✓
IncrementalAspect (Aspect)	✓	✓	✓
IncrementalDrainage (Watershed)	✓	✓	✓

```

2 dimensions of
  z, unit of -32768..32767
end

neighbor[] for north,south,east,west :=
  [0,1].z, [0,-1].z, [1,0].z, [-1,0].z

diff := 0
maxslope := 0
slope := 0

forall i:0..3
  diff := cell.z - neighbor[i]
  diff := diff * (-1) when diff < 0
  slope := diff * 100 / cell.unit
  maxslope := slope when slope >= maxslope
end

cell.z := maxslope
    
```

Fig. 8 Increment gradient (slope) operation

부터 자유롭게 구성될 수 있다. 여기에는 모두 6개의 연산들이 포함되는데, 이중에서 focal proximity (buffer), focal neighbor(Thiessen polygon), 그리고 focal spreading(cost surface) 연산만 구현되었다

(Table 4). 나머지 3개의 연산들은 알고리즘의 복잡성과 전역변수의 문제 등에 의해 본 연구에서는 구현되지 못하였다. Figure 9과 10은 각각 Thiessen polygon 연산과 비용거리(cost distance) 연산의 코드이다.

Tomlin의 마지막 연산 그룹인 구역 연산(zonal operations)은 각 셀이 포함된 구역에 의해 새로운 셀의 값이 계산되는 연산들로 근린 연산과는 달리 그 구역들의 형태와 크기가 매우 불규칙하다. 또한 구역들이 거리가 아닌 위상학적 관계(topology)에 의해 정의되기 때문에 이론적이나 기술적으로 CA 시뮬레이터로 구현하기가 매우 어렵다. 따라서 본 연구에서는 구역 연산은 구현되지 못하였다.

Table 4. Focal operations of extended neighborhoods

Focal operations defined by Tomlin (1990)	Focal operations in IDRISI	Focal operations implemented in the CA simulators	Focal operations of the integrated CA-GIS system
FocalProximity (Buffer)	✓	✓	✓
FocalNeighbor (Thiessen Polygon)	✓	✓	✓
FocalBearing			
FocalGravitation (Interpolation)	✓		✓
FocalRadiating (Viewshed)	✓		✓
FocalSpreading (Cost Surface)	✓	✓	✓

```

2 dimensions of
  seed of 0..30000
  barrier of 0..1
end

neighbor[] for n,e,s,w :=
  [0,1], [1,0], [0,-1], [-1,0]

forall i:0..3
  if (neighbor[i].seed > 0) & ([0,0].seed = 0) &
    ([0,0].barrier = 0) then
    cell.seed := neighbor[i].seed
  end
end
    
```

Fig. 9 Cellang code for Thiessen polygon operation

```

2 dimensions of
  cost of 0..32766
  target of 0..1
  friction of 0..32766
  barrier of 0..1
end

neighbor[] for 8 :=
  [0,1],[1,0],[0,-1],[-1,0],[1,1],[1,-1],[-1,-1],[-1,1]

forall i:0..7
  if (neighbor[i].target > 0 & cell.cost = 0 & cell.barrier = 0)
then
  if i < 4 then
    cell.cost := neighbor[i].cost + cell.friction
    cell.target := 1
  else
    cell.cost := neighbor[i].cost + 141 * cell.friction / 100
    cell.target := 1
  end
end
end
end

```

Fig. 10 Cellang code for cost distance operation

경사도 연산(slope operation)  
 사면 방향 연산(aspect operation)  
 Buffer operation  
 Thiessen polygon 연산  
 비용 거리 연산(cost distance operation)

성능 평가 시험은 여섯 종류의 크기가 다른 데이터(64 x 64, 128 x 128, 256 x 256, 512 x 512, 768 x 768, 1024 x 1024)를 사용하여 그 수행시간의 측정비교로 이루어졌다. 연산의 수행시간은 필자에 의해 작성된 프로그램을 사용하여 100분의 1 초 단위까지 측정되었고, 중앙처리장치(CPU)와 기억장치의 부하량은 실험중에 일정하게 유지되었다.

상수 연산은 Table 5에서 나타난 것처럼 데이터 크기에 상관없이 IDRISI가 월등히 나은 성능을 보여주고 있다. 또한 IDRISI의 ASCII 데이터 형태와 binary 데이터 형태를 비교해보면 binary 형태로 작성된 데이터가 더 빠른 속도로 처리됨을 알 수 있다. 중첩 연산도 상수 연산과 마찬가지로 대부분의 경우 IDRISI의 성능이 뛰어난 것으로 나타났다. 이 실험에서는 레이어 수도 변화시켜 보았는데, 대부분의 경우 IDRISI가 더 나은 성능을 지니고 있는 것으로 보여진다(Table 6와 7). 평균 필터의 경우는 네 번의 반복 계산의 경우까지는 IDRISI가 그 이후는 Cellular에 의해 구현된 평균 필터의 처리속도가 더 빠른 것으로 나타났다. 이 실험으로 반복적인 연산의 경우 CA 시뮬레이터가 반복 계산(iterative processing)에 효율적이고 우수한 성능을 지니고 있음이 증명되었다(Table 8과 9). Table 10와 11에서 나타난 것처럼, 경사도와 사면방향 연산은 모두 IDRISI가 더 뛰어난 성능을 가지고 있는 것으로 나타났다. Buffer 연산은 두 경우 (5 cell distance와 10 cell distance) 모두 Cellular가 더 나은 것으로 보이며 (Table 12과 13), Thiessen polygon의 경우 (Table 14) IDRISI가 더 나은 성능을 지니고 있음이 확인됐다. 마치

## 연산 성능 평가

지금까지 가온 GIS 연산을 CA 시뮬레이터를 사용하여 구현해 보았다. CA 시뮬레이터가 가진 한계와 전역변수의 문제로 인해 모든 GIS 연산들이 CA로 구현될 수는 없었다. 그러나 CA 시뮬레이터로 구현된 연산들 중 반복적인 프로세싱을 요구하는 연산들은 시뮬레이터가 가진 효율적인 처리 능력에 의해 원래 연산들 보다 우수한 성능을 가지는 것으로 나타났으며, 이를 실제로 계량적인 성능평가 시험에 의해 알아보았다. 그러나 이러한 일련의 평가시험은 CA 시뮬레이터에 의해 구현된 GIS 연산들이 코드의 최적화 과정을 거치지 않았기 때문에 상대적인 우위성만을 나타낸다.

국지 및 근린 연산을 대표하는 8개의 연산들이 이 평가실험에 선정되었다. 그 8개의 연산들은 아래와 같다.

- 상수 연산(scalar add operation)
- 중첩 연산(overlay add operation)
- 평균 필터 연산(low-pass filter)



막으로 비용거리 연산의 경우, 두 알고리즘 (cost push와 cost grow) (Eastman 1989) 모두가 평가에 사용되었는데, Cellular가 256 x 256 데이터 크기까지는 더 나은 성능을 보여주고 있고 그 보다 큰 데이터의 경우는 IDRISI의 cost push 알고리즘이 더 나은 성능을 보여주고 있다 (Table 15).

Table 5. Performance of scalar add operation  
(Min : Sec.Sec100)

DATA SIZE	IDRISI (ASCII FILE FORMAT)	IDRISI (BINARY FILE FORMAT)	CELLULAR
64 x 64	00:00.29	00:00.22	00:01.49
128 x 128	00:00.60	00:00.27	00:04.06
256 x 256	00:01.87	00:00.55	00:16.97
512 x 512	00:06.90	00:01.54	01:10.53
768 x 768	00:16.57	00:03.74	02:41.20
1024 x 1024	00:29.42	00:06.86	04:50.07

Note : The shaded values show the best performance in all tables.

Table 6. Performance of overlay add operation (IDRISI)

DATA SIZE	2 LAYERS	4 LAYERS	8 LAYERS	16 LAYERS
64 x 64	00:00.35	00:01.02	00:02.27	00:04.87
128 x 128	00:00.87	00:02.56	00:06.01	00:12.94
256 x 256	00:02.96	00:08.72	00:20.48	00:45.73
512 x 512	00:10.92	00:35.45	01:25.93	03:10.24
768 x 768	00:25.65	01:24.73	03:21.10	07:22.57
1024 x 1024	00:45.80	02:31.58	06:02.64	13:23.60

Table 7. Performance of overlay add operation (Cellular)

DATA SIZE	2 LAYERS	4 LAYERS	8 LAYERS	16 LAYERS
64 x 64	00:01.21	00:01.49	00:02.25	00:03.73
128 x 128	00:04.28	00:05.55	00:08.68	00:15.81
256 x 256	00:18.45	00:29.82	00:37.51	01:02.83
512 x 512	01:12.54	01:36.67	02:29.84	04:07.94
768 x 768	02:44.06	03:40.86	05:34.94	*
1024 x 1024	04:53.52	06:28.55	*	*

\* These cases were not applicable due to the limitation of Cellular.

Table 8. Performance of low pass filter (IDRISI)

DATA SIZE	1 TIME	2 TIMES	4 TIMES	8 TIMES	16 TIMES
64 x 64	00:00.44	00:00.66	00:00.87	00:01.94	00:04.04
128 x 128	00:00.99	00:01.54	00:02.18	00:04.96	00:10.63
256 x 256	00:02.58	00:05.15	00:07.34	00:16.56	00:35.85
512 x 512	00:09.55	00:18.78	00:28.39	01:04.88	02:18.67
768 x 768	00:21.86	00:44.32	01:06.17	02:32.93	05:25.79
1024x1024	00:39.44	01:19.91	01:59.51	04:36.35	09:43.34

Table 9. Performance of low pass filter (Cellular)

DATA SIZE	1 TIME	2 TIMES	4 TIMES	8 TIMES	16 TIMES
64 x 64	00:00.77	00:00.82	00:00.88	00:00.93	00:00.99
128 x 128	00:02.91	00:03.02	00:03.08	00:03.35	00:03.79
256 x 256	00:12.47	00:12.74	00:13.18	00:14.11	00:15.88
512 x 512	00:51.96	00:53.23	00:54.82	01:00.14	01:06.41
768 x 768	02:00.07	02:01.39	02:07.37	02:10.12	02:26.65
1024x1024	03:26.79	03:41.90	03:44.92	03:53.88	04:23.14

Table 10. Performance of slope operation

DATA SIZE	IDRISI	CELLULAR
64 x 64	00:00.61	00:01.18
128 x 128	00:01.90	00:04.56
256 x 256	00:07.36	00:18.87
512 x 512	00:29.25	01:15.01
768 x 768	01:00.03	02:44.77
1024 x 1024	01:46.86	04:49.40

Table 11. Performance of aspect operation

DATA SIZE	IDRISI	CELLULAR
64 x 64	00:00.86	00:01.13
128 x 128	00:02.42	00:03.57
256 x 256	00:09.04	00:15.25
512 x 512	00:37.68	00:58.19
768 x 768	01:18.65	02:09.95
1024 x 1024	02:21.16	03:49.76

Table 12. Performance of buffer operations (5 cells distance)

DATA SIZE	IDRISI	CELLULAR
64 x 64	00:02.66	00:00.82
128 x 128	00:09.86	00:02.75
256 x 256	00:38.11	00:11.48
512 x 512	02:31.73	00:46.30
768 x 768	05:46.25	01:43.29
1024 x 1024	10:30.49	03:03.73

Table 13. Performance of buffer operations (10 cells distance)

DATA SIZE	IDRISI	CELLULAR
64 x 64	00:02.72	00:01.02
128 x 128	00:09.85	00:02.91
256 x 256	00:38.04	00:12.11
512 x 512	02:31.51	00:49.08
768 x 768	05:45.42	01:52.85
1024 x 1024	10:28.23	03:23.11

Table 14. Performance of Thiessen polygon

DATA SIZE	IDRISI	CELLULAR (MIN. ITERATION)	CELLULAR (MAX. ITERATION)
64 x 64	00:00.44	00:01.54	00:02.26
128 x 128	00:01.18	00:07.20	00:13.66
256 x 256	00:03.87	00:42.93	01:32.00
512 x 512	00:14.61	04:41.25	11:25.19

Table 15. Performance of cost distance operation

DATA SIZE	IDRISI (COSTPUSH)	IDRISI (COSTGROW)	CELLULAR (MINIMUM ITERATION)	CELLULAR (MAXIMUM ITERATION)
64 x 64	00:05.33	02:31.14	00:02.34	00:04.23
128 x 128	00:20.66	*	00:12.97	00:26.21
256 x 256	01:22.31	*	01:21.13	03:10.92
512 x 512	05:27.79	*	09:24.63	24:45.26

\* These cases took more than an hour, so they are not measured.

## 새로운 GIS 연산의 구현

CA는 기존 GIS 연산들의 성능을 향상 시킬 뿐만 아니라 현재의 GIS에서 구현하기가 어려운 새로운 연산들을 구현할 수 있는 잠재력을 가지고 있다. 첫 번째로 CA를 이용하면 근린 연산에서 주변셀들을 자유사재로 정의할 수 있다는 장점이 있다. 따라서 다양한 형태, 크기 및 차원의 공간 필터를 쉽게 구성할 수 있다. 현재 GIS 중 소수의 시스템이 이러한 불규칙한 주변셀들을 구성할 수 있는 기능을 갖고 있지만, 다차원 혹은 시공간 필터 (spatio-temporal spatial filter)를 제공하는 시스

템은 아직까지 개발되어 있지 않다.

본 연구에서는 다양한 형태의 공간필터와 두 유형의 시계열 공간필터(time-series spatial filter)가 시험 제작되었다. CA에서 공간 필터를 제작하는 것은 그리 어렵지가 않다. Figure 11에서처럼 2 x 2 평균필터의 경우, 각 주변의 셀들은 중심(focus) 셀로부터의 상대적 위치에 의해 정의되고, 이 주변셀들에 적절한 가중치가 곱해지며 (이 경우 가중치는 1) 이들의 평균값이 중심셀의 값으로 저장된다. 예제로 3개의 비대칭(asymmetric) 평균 필터가 작성되었는데, 이들의 마스크는 2 x 2, 3 x 2, 4 x 4 였다. 또한 사각형태의 필터외에도 원형, 쉘기형태, 마름모 및 육각형의 평균필터가 시험 제작되었다.

2 dimensions of -32767..32766

neighbor [] for 4 :=

[0,0], [1,0], [0,-1], [1,-1]

weight [] for 4 :=

[0,0], [1,0], [0,-1], [1,-1]

forall i:0..3

weight[i] := 1

end

sum := 0

forall i:0..3

sum := sum + (neighbor[i] \* weight [i])

end

cell := sum / 4

Fig. 11 Cellang code for the 2 x 2 mean filter

CA 시뮬레이터에서 개개의 셀은 하나 이상의 상태(local states)를 가질수 있다. 특히 Cellular는 이론적으로 256개의 다른 셀 상태를 가질 수 있어, 이를 이용하면 시계열 필터를 만들어 낼 수 있다. 이러한 다중 상태 구조는 래스터 GIS에서의

다중 레이어 개념과 일치하여 시계열 데이터가 복수 셀 상태로 셀틀라 공간에서 구성될 수 있다. 첫 번째로 국지 시계열 필터(local time-series filter)가 시험 제작되었다. 이는 시계열 데이터를 다중 셀 상태의 구조로 구성한 다음, 적절한 가중치를 셀 상태에 부여하여 데이터를 평활화(smoothing)한다 (Figure 12). 국지 시계열 필터는 또한 공간 차원으로도 확대될 수 있다. 근린 시계열 필터(focal time-series filter)는 국지 시계열 필터에서 주변셀들을 공간차원으로 확대하여 가중치를 모두 시간과 공간차원에 적용하여 평활화한다(Figure 13). 따라서 이 근린 시계열 필터는 시공적으로 셀의 값을 처리하여 다차원 평활화 효과를 낼 수 있다.

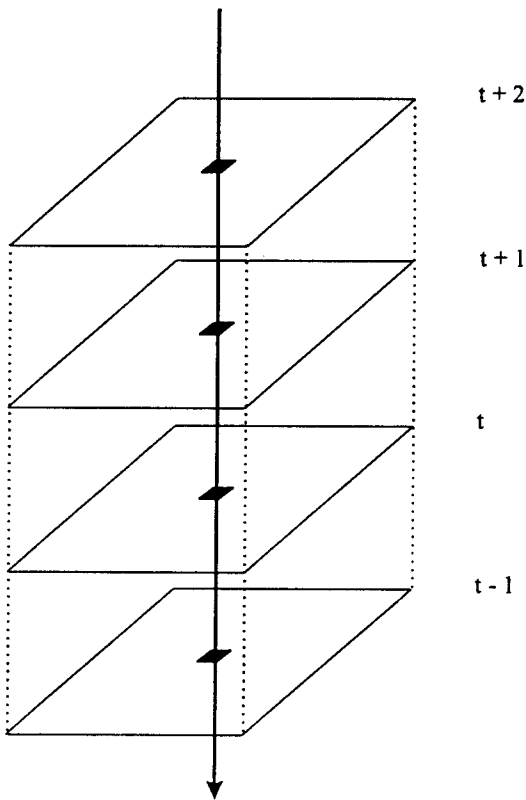


Fig. 12 Local time-series filter

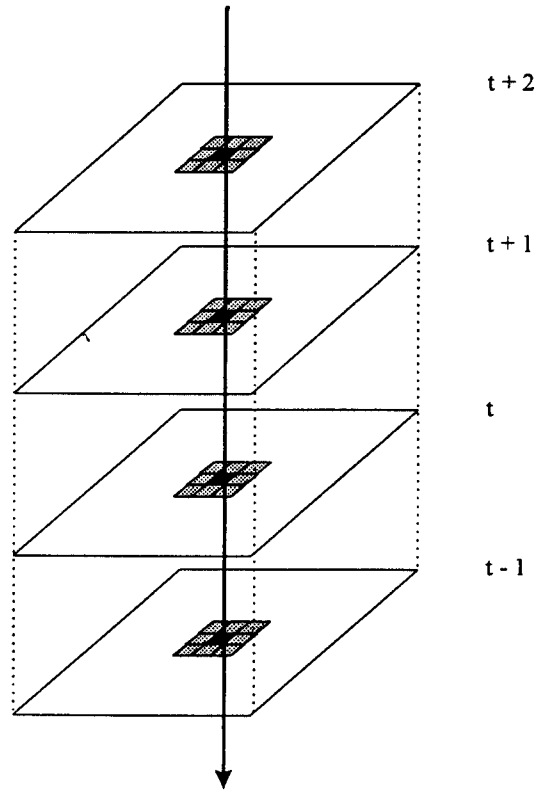


Fig. 13 Focal time-series filter(3 × 3)

앞에서 소개된 필터 외에도 CA 시뮬레이터를 이용하면 다차원 공간 필터를 쉽게 제작할 수 있다. 이들 필터는 근린 시계열 필터와 개념상 유사하다. 다차원 공간 필터 제작에서의 중요한 요소는 다차원 근린(혹은 마스크)의 구성이다. 예를 들어 3 x 3 평균 필터는 3차원 공간으로 확대되어 3 x 3 x 3의 voxel 마스크가 사용된다. 이것은 가중치가 x와 y축 뿐만 아니라 제 3차원에도 적용될 수 있음을 의미한다. 따라서 만약에 제 3차원이 시간차원이면 이 필터는 시공 데이터에 적용될 수 있는 근린(시공)시계열 필터가 된다. 더 나아가 다차원 필터는 보다 높은 차원 (즉, 4차원 혹은 5차원)에도 쉽게 적용될 수 있다.

또 다른 유형의 필터로는 동적필터이다. 이는

## 결론

각 시간별로 다른 필터가 적용될 수 있음을 의미한다. 하나의 예로써 Figure 14에 보여진 가스 확산 연산자를 들 수 있다. 이 연산자는 Margolus neighborhood (Toffoli and Margolus 1987)를 사용하고 있는데, 이 주변셀의 구성이 매시간 시계방향 혹은 반시계방향으로 회전하여 결과적으로 입자나 물체가 무작위하게 움직일 수 있게 해준다. 이와같은 동적필터들은 가스확산이나 거주지의 확산, 인구이동 등 다양한 확산현상의 연구에 도움을 줄 것으로 생각된다.

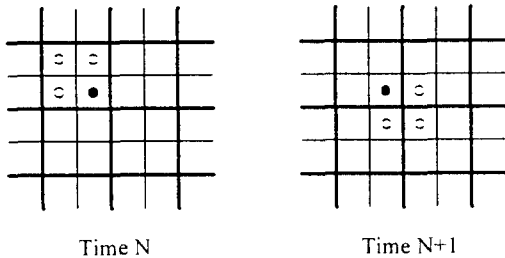


Fig. 14 Margolus Neighborhood (After Toffoli and Margolus 1987)

마지막으로 CA 시뮬레이터를 이용하여 지리적 확산 연구에 도움을 줄 공간 확산자를 개발 할 수 있다. CA는 이러한 공간 확산의 연구에 필요한 도구와 방법론을 제공하고 있으며 또한 이미 많은 유형의 공간 확산자가 개발되어 왔다. 그 하나의 예로, diffusion-limited aggregation (DLA)를 들 수가 있다. DLA 모델은 원래 금속 및 먼지입자 등과 같은 작은 입자가 모여 큰 결정체를 구성하는 과정을 모의실험하기 위해 개발된 모델로 (Witten and Sander 1981, 1983), 지리학에서는 도시성장의 기하학적 특성을 연구하기 위해 사용되어 왔다(Batty et al., 1989; Batty 1991; Batty and Longley 1994). 이러한 DLA 모델들은 CA 시뮬레이터를 이용하여 구현될 수 있으며(Toffoli and Margolus 1987), 이론적인 수준에서의 모의 실험을 가능하게 해준다.

본 연구는 CA 시뮬레이터가 GIS의 분석기관으로 결합되어 있는 통합 CA-GIS를 이용하여 기존의 GIS 연산들을 구현하고 연산들의 성능을 향상시키며, 현재의 GIS에서는 구현이 불가능하거나 어려웠던 연산들을 새로이 개발하는데 그 목적을 두었다.

대다수의 국지 및 근린 GIS 연산들이 통합 CA-GIS에 의해 성공적으로 구현될 수 있었으며, 이 중 반복적인 프로세싱을 요구하는 일부 연산들은 CA가 갖고 있는 우수성에 기인해 원래 연산보다 더 나은 성능을 보여주었다. 또한 새로운 유형의 연산들 (다양한 형태의 공간필터, 시계열 필터, 확산 연산자 등)이 시험 제작 되었고 다차원 공간 연산, 시공연산 그리고 동적 연산들이 이 통합 시스템을 사용하여 구현될 수 있음이 예시되었다.

그러나 CA와 GIS의 이론적 차이와 시뮬레이터의 한계에 의해 통합 CA-GIS 시스템은 모든 GIS 연산들을 구현할 수 없었다. 그 이유로 첫 번째 전역 변수의 문제를 들 수 있다. 모든 CA는 국지 프로세싱에 기초하고 있다. 즉, 각 셀의 상태는 인접한 주변 셀들의 상태에 의해서만 정의된다. 따라서 CA에서 전역 변수를 설정하기가 매우 어렵고 결과적으로 알고리즘상 전역 변수를 사용하는 일부의 GIS 연산들이 본 연구에서 구현될 수 없었다. 두 번째의 한계는 CA 시뮬레이터들이 일반 프로그래밍 언어나 GIS와는 달리 사용자로부터의 다양한 유형의 인수(parameters)들을 받아 들이기 어렵다는 점이다. 이는 재분류(reclassification) 연산이 CA 시뮬레이터로 구현되기 어렵다는 점에서 잘 나타나고 있다.

CA는 갖고 있는 여러 가지의 우수한 특성 때문에 현재 GIS가 해결해야 할 여러 가지의 과제들 즉, 연산기능의 확장, 성능의 향상 그리고 새로운 연산의 제작 및 시험에 이론적, 실제적 이점을 제

공해주리라 기대된다. 그러나 CA가 자료의 입력, 저장, 검색 및 출력 등의 GIS 기능들을 완전히 대체할 수는 없으며 또 GIS에서 CA처럼 효율적이고 능률적인 반복연산, 시공처리, 그리고 다차원 연산 능력을 기대하기는 어렵다. 따라서 앞서 언급된 GIS의 문제들을 해결하는데 있어 최선의 방법은 CA 시뮬레이터를 GIS의 분석 기관으로 사용하여 두 시스템이 가진 장점들을 모두 활용하는 것이다.

## 참 고 문 헌

- Batty, M., P. Longley, and S. Fotheringham, 1989, "Urban growth and form : scaling, fractal geometry, and diffusion-limited aggregation", *Environment and Planning A*, Vol. 21, pp.1447-1472.
- Batty, M., 1991, "Generating urban forms from diffusive growth", *Environment and Planning A*, Vol. 23, pp.511-544.
- Batty, M. and P. Longley, 1994, *Fractal Cities*, Academic Press, London.
- Batty, M. and Y. Xie, 1994, "From cells to cities", *Environment and Planning B : Planning and Design*, Vol. 21, pp.s31-s48.
- Clarke, K. C., J. A. Brass, and P. J. Riggan, 1994, "A cellular automaton model of wild fire propagation and extinction", *Photogrammetric Engineering and Remote Sensing*, Vol. LX, No. 11, pp.1355-1367.
- Couclelis, H., 1985, "Cellular worlds : a framework for modeling micro-macro dynamics", *Environment and Planning A*, Vol. 17, pp.585-596.
- Couclelis, H., 1987, "Cellular dynamics : How individual decisions lead to global urban change", *European Journal of Operational Research*, Vol. 30, pp.344-346.
- Couclelis, H., 1988, "Of mice and men : what rodent populations can teach us about complex spatial dynamics", *Environment and Planning A*, Vol. 20, pp.99-109.
- Deadman, P., R. D. Brown, and H. R. Gimblett, 1993, "Modeling rural residential settlement patterns with cellular automata", *Journal of Environmental Management*, Vol. 37, pp. 147-160.
- Eastman, J. R. 1989, "Pushbroom algorithms for calculating distances in raster grids", *Proceedings, Ninth International Symposium on Computer-Assisted Cartography, AutoCarto 9*, Baltimore, Maryland.
- Eastman, J. R., 1992, *IDRISI User's Guide*, Clark University, Graduate School of Geography, Worcester, Massachusetts, 01610.
- Eckart, J. D., 1995, "A Cellular Automata Simulation System", Computer Science Department, Radford University.
- Gardner, M., 1971, "On cellular automata, self-reproduction, the garden of eden and the game life", *Scientific American*, Vol. 224, No.2, pp.112-117.
- Green, D. J., 1982, "Simulated effects of fire, dispersal and spatial pattern on competition within forest mosaics", *Vegetation*, Vol. 82, pp.139-154.
- Green, D. G., A. P. N. House and S. M. House, 1985, "Simulating spatial patterns in forest ecosystems", *Mathematical and Computers in Simulation*, Vol. 27, pp. 191-198.
- Gutowitz, H., 1991, *Cellular Automata : Theory*

- and Experiment*, The MIT Press, Cambridge, Massachusetts.
- Itami, R. M., 1988, "Cellular automata as a framework for dynamic simulations in geographic information systems", *GIS/LIS 88 Proceedings*, pp.590-507.
- Itami, R. M. and J. D. Clark, 1992, "Spatial dynamic simulations using discrete time and discrete event theory in cell based GIS systems", *The 5<sup>th</sup> International Symposium on Spatial Data Handling*, Charleston, South Carolina, pp. 702-712.
- Kirtland, D., L. DeCola, L. Gaydos, W. Acevedo, K. Clarke, and C. Bell, 1994, "An analysis of human-induced land transformations in the San Francisco Bay/Sacramento area", *World Resource Review*, Vol. 6, No. 2, pp.206-217.
- Park, S., 1996, "Integration of Cellular Automata and Geographic Information Systems for Modeling Spatial Dynamics", Unpublished Ph. D. Dissertation, Department of Geography, University of South Carolina.
- Silvertown, J., S. Holtier, J. Johnson, and P. Dale, 1992, "Cellular automaton models of interspecific competition for space - the effect of pattern on process", *Journal of Ecology*, Vol. 80, pp.527-534.
- Tobler, W. R., 1979, "Cellular Geography" in Gale and Olsson (eds.), *Philosophy in Geography*, Reidel Publishing Company, Dordrecht, Holland, pp.379-386.
- Toffoli, T. and N. Margolus, 1987, *Cellular Automata Machines : A New Environment for Modeling*, The MIT Press, Cambridge, Massachusetts.
- Tomlin, C. D., 1990, *Geographic Information Systems and Cartographic Modeling*, Prentice Hall, Englewood Cliffs, New Jersey.
- Von Neuman, John, 1966, *Theory of Self-Reproducing Automata*, University of Illinois Press.
- Wagner, D. F., 1994, "Cellular automata and geographic information systems", American Association of Geographers Presentation Paper, San Francisco.
- Wagner, D. F., 1997, "Cellular automata and geographic information systems", *Environment and Planning B* (in press).
- Wilson, G., 1987, "The life and times of cellular automata", *New Scientist*, Vol. 8, pp. 44-47.
- Witten, T. A. and Sander L. M., 1981, "Diffusion-limited aggregation, a kinetic critical phenomenon", *Physical Review Letters*, Vol. 47, No. 19, pp.1400-1403.
- Witten, T. A. and Sander L. M., 1983, "Diffusion-limited aggregation", *Physical Review B*, Vol. 27, No. 9, pp. 5686-5697.
- White, R. and G. Engelen, 1993, "Cellular automata and fractal urban form : a cellular modeling approach to the evolution of urban land-use patterns", *Environment and Planning A*, Vol. 25, pp.1175-1199.
- White, R. and G. Engelen, 1994, "Cellular dynamics and GIS : modeling spatial complexity", *Geographical Systems*, Vol. 1, pp.237-253.
- Wolfram, S., 1984a, "Preface", *Physica 10D*, pp.vii-xii, Elsevier Science Publisher, North-Holland, Amsterdam.

Wolfram, S., 1984b, "Universality and complexity in cellular automata", *Physica 10D*, pp.1-35, Elsevier Science Publisher, North-Holland, Amsterdam.