調査論文

문자 패턴의 크기변환

正會員 李 壽 淵*

Size Transformation of Character Pattern

Soo Youn LEE*, *Regular Member*

요 약 한글Wordprocessor, Workstation 및 Telematics용 I/O Device에서는 각종 크기의 문자나 도형등을 요구하고 있다. 본 논문에서는 내장된 문자 dot pattern을 축소 또는 확대시키기 위하여 지금까지 연구되어온 각종 방식의 알고리즘을 논한다. 또한 문자 dot pattern의 크기를 변환시키는 방법을 평가하기 위한 항목을 논하며 이항목에 따라서 종래의 방식을 주관적으로 평가한다.

ABSTRACT Hangeul wordprocessor, workstation and I/O device for telematic service are requiring many kinds of characters with various attributes such as size and font. In this paper, we survey computer algorithms that have been studied to transform(reduction, enlargement) built-in character patterns into character patterns with different sizes. Also, evaluation criteria for such algorithms are mentioned.

1. 서 론

요즘 Office automation 및 문서정보 통신의 발전에 힘입어 일반문서 작성용 work station (이하 WS)에서는 종래의 문자뿐만 아니라, 도형 화상을 용이하게 처리 또는 편집하는 기능이 요구되고 있으며, 일본등과 같은 선진국에서는 자국어 처리능력에도 도형처리, 계산기능력을 부가한 복합 WS가 이미 개발되어 발표되고 있다.

이와같은 WS에서 화상데이터를 자유로이 편집을 행하기 위하여(기억장치에 저장되어 있는 화상 데이터를 작성문서에 알맞게 배치하기 위하여) 화상 데이터의 축소, 확대, 이동등의 기능이 필수적이다.

* 光云大學電子計算機工學科
Dept. Computer Engineering Kwangwoon Universty, Seoul, 132 Korea.
論文番號 : 86-09(接受 1986. 4. 2)

지금까지의 연구로서는 기억장치속에 내장된 문자(주로 한자)를 여러가지 크기로 변환 표시하거나 또는 해상도가 상이한 display상에 표시하기 위한 연구가 후술하는 바와같은 각종 방식이 연구되어 왔다.

종래의 연구는 한자를 대상으로한 변환이 그 주류가 되어 왔으며 그 변환 처리의 기본단위도 주로 화소(pixel) 단위를 채택하여 왔다. 그러나 문자, 도형을 포함하는 선도형을 정보압축, 변환, 전송등의 목적에 따라 관측하거나 처리하는 단위로써 화소, 선소 및 면을 들수 있다. 화소단위의 처리는 변환방식이 비교적 간단하며 변환패턴의 품질을 유지하는 것이 어려운 반면에 선분단위의 변환은 처리방식이 복잡해지는 반면 선소단위는 화소단위의 국소적 표현보다 선도형을 보다 충실히 표현할 수 있으므로 변환패턴의 품질이 양호하게 되는것이 일반적이다.

이후 문자패턴의 축소 및 확대에 따라서 고려되어야 할 일반적 사항을 논하며, 1970년도 중

반부터 1986년까지 연구되어온 각종방식, 결론의 순으로 논한다.

2. 문자 패턴 변환을 위한 고려사항

처리 대상인 2치 패턴(dot pattern 이라고도 함)을 원 패턴이라 하며 m행 * n열의 matrix C로 표현한다.

원 패턴C는 확대 또는 축소변환(이하 차수변환이라고 함)에 의해 M형 * N열의 dot pattern C^T로 변환이 되는 C^T를 변환 패턴이라 부르며 $Rx=M/m$, $Ry=N/n$을 각각 종변환비율, 횡변환비율이라 정의한다. 변환비율은 변환차수라고도 한다.

$$C=\{a(i,j)\} \quad i=1-m,\ j=1-n$$
$$C=\{a(I,J)\} \quad I=1-M,\ J=1-N$$
$$a(i,j)=\begin{cases}0 : \text{백 Dot}\\ 1 : \text{흑 Dot}\end{cases}$$

기타 필요한 용어 정의는 해당 하는 곳에서 필요에 따라 설명키로 한다.

1) **처리 대상** - 한자 dot 패턴의 변환과 같이 비교적 작은 size의 패턴 처리와(예, 명조체, 24 * 24 dot크기)문자열 또는 문자, 도형이 섞여 있는 복합 패턴과 같은 비교적 큰 size 의 패턴처리로 대별되며, 처리대상의 범위에 따라 응용범위도 상이하게 된다. 일반적으로 종래의 방식은 전자에 국한되어 왔으나 장차는 문서작성시 고도의 편집기능을 이용하여 문자뿐만 아니라, 도형이나 도면의 패턴변환까지 고려되어야 할 것이다. 궁극적으로는현재와 같은 2치패턴을 포함하여, 농담화상 및 컬러화상의 크기변환도 고려해야할 것이다.

2) **변환 방법** - 변환은 축소, 확대, 회전을 들수가 있다. 종래의 방법은 횡, 종 방향 다같이 축소 또는 확대에만 치중되어 왔으나 횡방향, 종방향의 확대, 축소가 각각 가능하며 또한 고품질의 문서화상을 작성하기 위하여 회전처리도 고려되어야 한다. 물론 회전처리에서 발생되는 변환패턴 품질의 열화는 매우 클것으로 예상되므로, 회전처리에 따른 품질유지도 적극적으로 고려되어야 한다.

3) **변화 비율** - 정수배의 비율과 J/I 실수배 비율의 축소, 확대로 나누어진다. 원 패턴의 변환 기술의 발전에 힘입어 변환비율도 정수배에서 실수배로 발전되어 왔으나 0.5배 - 2 배 정도의 실수배 변환이 일반적인 추세이다. 그러나 변환비율이 2 배이상일 때 변환 패턴의 품질열화가 현저해지는 문제점의 완화를 전제로 0.5배 - N배의 실수배 비율 변환방법도 고려되어야 한다.

4) **변화 기준** - 변환 패턴의 품질을 평가하는 기준으로서 a) 직선선분의 stroke폭 유지, b) 패턴의 균형유지, c) 사선 선분처리의 품질유지 등을 고려하여야 한다. 일반적 추세로서는 a) 보다는 b), c)를 중심으로 처리하는 경향이 많다. 그러나 변환 비율의 확대에 따른 선도형의 stroke폭 제어도 변환패턴의 품질에 많은 영향을 끼치므로 확대비율에 적응하는 stroke폭 제어도 고려되어야 한다. 또한 변환처리에 따른 문자의 font속성(예 명조체) 의 유지 및 font속성의 변환(예 명조체에서 고딕체로 변환)에 따른 처리 및 그 기준도 고려되어야 한다.

5) **품질 개선** - 4) 의 변환 기준에 따라각기의 방법상 기준에 의거 변환 처리한 후 후속단계로써 smoothing을 통하여 품질개선처리를 하여 왔으나 처리 알고리즘이 국소적이거나, 또는 너무 복잡하여 품질개선의 효과가 크지않은 경향이다. 따라서 원 dot pattern이 analog문자에 비하면 이미 품질의 열화가 발생되어 있으므로 원 패턴을 기준으로 변환 패턴을 품질 개선하는 것도 고려되어야 할 것이다. 즉, 후속적인 품질 개선을 전제로 하지 않는 차수 변환도 고려되어야 한다.

6) **변환 속도** - 차수변환의 속도를 결정하는데는 응용되는 분야와 비율, 패턴의 크기등에 좌우된다. 일반적으로 말하면 변환차수가 커질수록 처리속도는 늦어지는 방법이 많다. 여기서 응용면이라함은 단순한 문자를 여러 크기로 발생하기 위한 것이나, 또는 고성능 WS

에 사용되는 것등을 말하며 일반적으로는 속도와 품질은 반비례하는 경향이 있다.

7) 기 타-원 패턴의 축소 및 확대 처리를 상대적으로 비교하여 보면 원 패턴을 저장하는 기억용량의 효율화를 추구하기 위해서는 원 패턴을 작은 크기로 작성하게 되어 결과적으로는 확대 변환처리가 많게 된다. 그러나 변환처리 결과, 품질 위주로 할 경우 원 패턴의 크기는 커지는 것이 당연하며, 이에 따라서 축소변환처리가 많이 일어난다고 본다. 일반적인 dot처리 기준의 차수변환에 있어서는 원 패턴의 크기에 비례하여 기억용량이 늘어나지만 원 패턴에 포함된 문자, 도형의 복잡도에는 영향을 받지 않는다. 그러나 방식에 따라서는(4방향 선분분리법, Vector법) 원 패턴의 크기에 따라 원 패턴의 기억용량은 비교적 영향을 받지 않으나, 원 패턴의 복잡도에 영향을 받게 되므로 고 품질의 변환처리를 위해서는 원 패턴의 크기가 클수록 유리하리라 판단된다.

3. 각종 방식

3-1 DOT MAPPING법

본 방식은 원 패턴의 흑 dot로 부터 변환 패턴의 흑 dot 또는 그 집합으로 변환시키는 방법(이하 forward dot mapping; FDM)과 변환 패턴의 각 dot로부터 원 패턴을 mapping시켜 그 주위 값에 따라 보간하는 방식(보통 보간처리라 하는 여기서는 후술하는 확대보간법과 혼돈을 피하기 위하여 이하 backward dot mapping; BDM이라함)으로 대별할 수 있다. 전자는 가장 초기에 연구된 방식이며 후자는 비교적 최근에 연구발표된 것이다.

이하 각각의 장단점에 대하여 약술한다.

1) FDM

일명 비례법이라고 하는데 원 패턴의 흑 dot a(i, j)를 다음식 조합으로 변환 패턴의 dot위치를 좌표(I, J)에 비례 배치하는 방법이다.

$$I1 = [Rx * i + 0.5] \cdots\cdots (1)$$
$$J1 = [Ry * j + 0.5] \cdots\cdots (2)$$
$$I2 = [Rx * (i+1) + 0.5] \cdots\cdots (3)$$
$$J2 = [Rj * (j+1) + 0.5] \cdots\cdots (4)$$

Rx, Ry는 확대비율, [ ]는 가우스 기호이다. 축소의 경우 좌표는 (I1, J1)이나 확대의 경우 (I1, J1), (I1, J2), (I2, J1), (I2, J2)와 같이 2점-4점까지 좌표를 발생시키는 방법으로 변환이 가장 간단하고 문자의 balance가 유지되는 장점이 있다. 단점으로는 문자의 폭이 변화되는 점을 들 수가 있고 또한 사선부분에서 凹凸이 많이 생기는 문제점이 있다.

변환 비율의 범위는 0.5-2배까지나 2배이상의 변환시 변환된 패턴을 이용하여 재변환 시켜야 한다. 본 방식의 처리 속도는 원 패턴의 크기와 변환 비율에 비례하여 내장된 문자 dot pattern을 변환시킬 때 처리속도는 매우 빠르나 문자품질이 좋지 못하여 열화가 심해 고품위 문서작성용으로는 실용적인 방법이라 할 수 없다.

2) BDM

원 패턴을 확대, 축소 시키기 위하여 우선 원 패턴위에 확대 또는 축소 비율의 역수 간격을 갖는 mesh를 덮어 씌우고 그 격자점을 변환 패턴의 1화소로 하는 방식으로, 변환 패턴의 격자점 Q좌표(I, J)로 부터 원 패턴상의 P점좌표(i,j)의 계산은 다음과 같다.

$$i = 1 + (m-1) / (M-1) * (I-1), \ (I = 1, 2, \cdots M)$$
$$j = 1 + (n-1) / (N-1) * (J-1), \ (J = 1, 2, \cdots N)$$

P점의 좌표(i, j)는 일반적으로 원 패턴의 격자점에 일치하지 않으므로 P점 주변에는 4개의 격자점(Pi; i=1, 2, 3, 4)이 존재하게 된다. 즉 본 방식은 Q점(I, J)의 값을 P점 주변의 4점의 관계로 부터 구하는 방식으로 4격자점의 영향에 따라 곡면보간법, 논리화법, 최근화법, 9분할법, 투영법, 거리반비례법, 16분할 table법 등을 들

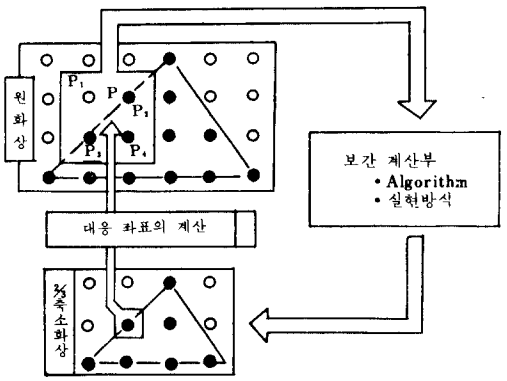

그림 1 보간계산
Interpolation.

수가 있다.

곡면보간법은 P점 주변에 있는 4개의 데이타점을 통하여 최저차의 곡면방정식으로부터 P점에서의 Z값(농도치)를 근사적으로 보간한 다음 실험을 통하여 정해진 threshold 값과 비교하여 흑 dot나 백 dot를 정한다.

논리화법은 P점 주위의 P1-P4 화소중 적어도 한점이 흑 dot이면 P의 값을 흑 dot로 하는 방법이며 최근방법은 P에 가장 가까운 주변점 Pi의 값을 P값으로 하는 방법이다.

9분할법은 P1-P4로 이루어지는 정방영역을 적절히 가로, 세로 3가지씩 나눈 결과 생기는 9개의 영역에서P점이 어느 영역에 속하느냐에 따라, 이미 설정된 각 영역과 P1-P4의 영향 관계로 부터 P점의 값을 구하게 된다.

16분할 table법은 P1-P4 4점간 영역을 16분할한 다음, 그림 2와 같은 16분할 table 패턴에 따라 P점의 값을 정하는 방법이다. 먼저 P1-P4점의 관계로 부터 16분할 패턴 중 한 패턴을 선택한 다음, 그 패턴내에서 P가 어느 영역에 속하느냐에 따라 P점의 값을 정하는 방법이다.

P가 그림 2의 패턴 중 사선으로 표시한 영역에 속하면 흑 dot로 한다.

본 방식의 공통적인 특성으로서 문자의 balance는 잘 유지되나, stroke의 폭이 변화하는 점은 FDM과 같다. 그러나 변환 패턴의 품질은 FDM보다 양호하다. 변환 비율은 J/I배로, 정수

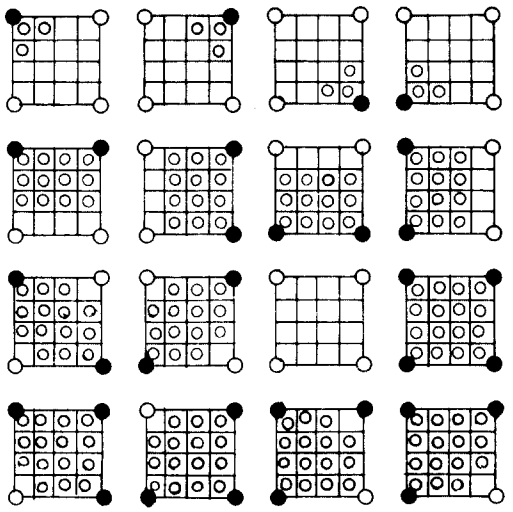

그림 2 4점사이 16분할 pattern 예
Table pattern example of 16 segmented area.

배, 비정수배로 원칙적으로는 변환 범위에(특히 확대시) 제한이 없으며 변환처리도 변환 패턴의 크기에 비례하는 특성을 갖는다. 그러나 threshold값에 따라 선폭이 변화되어 threshold 값의 선택에 유의하여야 한다. (경우에 따라서 1선분 stroke가 없어질수가 있다. 사선 부분이나, 직선 교차점 부근에서 불필요한 Dot의 발생으로 품질이 저하되므로 5치변환법에 의한 품질개선처리를 적용시켜 품질을 개선하나 처리시간이 길어진다. 특히 본 방식은 변환패턴의 크기에 차수 변환 시간이 길어지므로 확대비율이 큰경우 품질개선을 위한 처리시간은 매우 길어지게 된다. 그러나 16분할 table법은 차수변환을 부분적으로 hardware화 시킬수 있는 방법을 제시하였다는 점에서 주목할 필요가 있다. (다치 화상으로 확대 가능)

3-2 확대 보간법

Dot mapping법은 처리 단위를 dot단위로, 행열 선택법은 행 또는 열을 처리단위로 함으로써 사선의 품질에 많은 문제점이 발생하였다.

본 방식은 정형처리를 하지 않고도 사선의 품질을 유지시키기 위하여 고안된 방식이다. 본 방식의 처리 단위는 2 * 2, 3 * 3 등과 같은 mesh이다.

원 패턴 및 변환 패턴의 mesh크기는 변환비율에 따라 결정된다. 예를들면 원 패턴을 1.5배(3/2배) 확대할 경우 원 패턴의 mesh는 2*2 크기이며 변환 패턴의 mesh크기는 3*3이 된다. 원 패턴의 mesh종류는 $2^4=16$종이며, 변환패턴 mesh종류는 $2^9=512$종이 된다. 그러나 실제 문서화상에서 존재하는 각 패턴에서 mesh의 종류는 mesh크기에 따라 다소 차이는 있으나 상당히 감소하게 된다. 3/2확대시 원 패턴의 mesh 한 종류에 대하여 변환 패턴을 구성하는 mesh는 여러가지 존재가능하기에 1 : 다 대응이 되므로 이러한 방법은 실제적으로 응용하기 어렵다. 따라서 그림 3과 같이 2*2 mesh 의 각각에 대해 3*3 mesh를 1 : 1 대응시켜 변환시키는 방법이다. 물론 4/3 확대일경우 원 패턴의 mesh 크기는 3*3이며 변환 패턴을 구성하는 4*4 크기의 mesh를 1 : 1 대응시켜도 $2^9=512$종류가 된다. 이와 같은 이유로 본 방식의 확대비율을 분수로 표현할 때에만 가능하게 된다. 일단 분수형식으로 표현된다고 하나, 원 패턴의 처리단위인 mesh의 크기는 2*2 및 3*3정도이다.

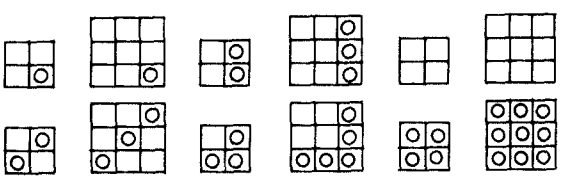

그림 3 3/2배 확대용 변환 Table의 일부
A part of transformation table.

본 방식에서 사선의 품질은 타방식과 비교할 때 고품질이나, 변환비율이 고정되어 실수배 변환처리가 어렵다. 또한 변환비율에 따른변환 mesh를 각각 준비(예로서, x/3배일때 512종, y/2배일때 16종등)하여야 하는 단점이 있어 실제 응용이 매우 어렵다. 물론 축소의 경우 mesh 의 대응은 2 : 1이다.

3-3 행열 선택법

본 방식은 축소, 확대를 행·열단위로 행하는 방법으로 문자의 균형, 또는 선분폭의 균일성등과 같은 목적에 따라 행·열을 적당히 선택하여 처리하는 방법이다. 즉, 축소의 경우 선택된 행, 또는 열을 그대로 제거하거나 인접행, 또는 열과 논리화를 시킨 다음 제거 하며 확대의 경우 선택된 행 또는 열을 중복시켜 출력하는 방식이다. 변환행(열)의 선택은 변환비율에 따라 기계적으로 선택하는 방법과 일단 기계적으로 변환행(열)을 선택한 다음, 그 행(열)의 주변상황을 고려하여 변환행(열)을 1행 또는 2행등의 범위내에서 다시 조정해(우선순위에 따라) 선택하는 방법이 있다. 전자의 방식은 문자의 국소적 특성(예, 1선폭의 선분)을 고려하지 않으므로, 축소 변환시 문자구조가 파괴되는 단점이 있고, 선분폭의 불균일성이 눈에 띄게된다. 이러한 단점을 보완하기 위해 후자의 방식이 도입되었다. 행(열)의 혹 dot분포 변화분포등을 이용하여, 축소·확대에 따른 영향이 적게 미치는 방향으로 변환행(열)을 국소적으로 조정선택한다. 본 방식은 FDM에 비해 처리단위가 틀린점 이외에는 변환품질등은 비슷하나 처리속도가 다소 개선된다.

3-4 선분 비례 배치법

선폭의 균일성을 유지시키기 위하여 문자의 balance를 눈에 띄지 않을 정도로 열화시키는 방법으로 행 또는 열 단위의 처리이지만 패턴 (문자) 수평 또는 수직 선분(stroke) 이라는 문자패턴 국소적인 성질을 이용하고 있다. 구체적으로 선분을 추출하기 위하여 행(열)단위의 주변분포, 변화분포, 최대연속 혹 dot 분포등의 통계적수법을 채택하였으며, 일단 추출된 선분을 포함한 행(열)을 비례 배치한 다음, 행간 또는 열간, 부분적으로 mapping시키는 방법이다.

본 방식은 임의의 변환 배율에 적용 가능하며 문자 balance도 비교적 유지되며 선폭도 유지되는 방식이나 처리가 너무 복잡하고 처리속도가 늦다는 결점이 있다. 본 방식의 중요한 특성으로서 축소 변환시 서로 떨어져 있는 복수개의 직선 선분이 연결되어 하나의 선분이 되어 문자의 본질적인 구조가 파괴되는 점을 방지한데에 주목할 가치가 있다.

3-5 직선 선분 추출법

본 방식은 원래 FAX, FSS등과 같은 장치로 입력된 한자 모형을 계산기와 작성자가 회화처리를 통하여 정형하여 한자 Dot 패턴을 작성하는데 이용되었다. 즉 한자 dot pattern중 종·횡 직선을 분리하여 선폭을 규격화시킴과 동시에, 종횡직선에 수반하는 잡음을 제거하여 품질을 개선 하였다.

차수변환처리는 분리시킨 종·횡직선과 나머지 부분(사선부분으로 판정)을 소정의 크기 matrix로 mapping시킨다. 본 방식에서 추출된 직선은 선폭이 1이상이며 일정 길이 이상을 갖는 직선의 추출로써 한자 dot pattern이 갖는 구조적인 성격을 파악 이용한다는 점이 평가되어야 한다.

선분 비례 배치법에서는 행, 또는 열단위의 선분 구성요소를 파악하고 기계적 변환처리에 비하여 좀더 고수준의 처리라 할 수 있다. 사선 부분의 처리는 dot mapping방식으로 생각된다.

3-6 VECTOR 표현법

전술한 방법은 처리대상을 한자 dot pattern 처리를 주로 하였으며 변환도 종·횡 방향의 축소, 확대를 중심으로 연구되어져 왔다. 그러나

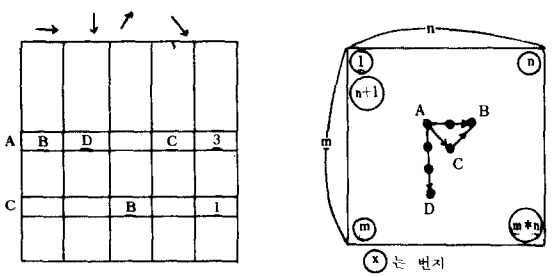

그림 4 처리 예
Example of Processing.

본 방식은 한자·화상·도형을 구분치 않고 표시시킴과 동시에 종·횡방향의 각각 축소, 확대를 추구하며 또한 회전도 자유롭게 시키기 위하여 연구된 방식이다.

이 방식은 원 패턴을 차수변환시키기 앞서 원 패턴에 속하는 모든 흑 dot에 대하여 각 흑 dot에서 출발하는 종방향, 횡방향, 45°방향, −45° 방향과 같은 4방향선분에 속하는 흑 dot중 각 방향에서 제일 끝에 있는 흑 dot를 관련시키게 테이블 형태로 표현된다. 원 패턴이 m*n 크기인 경우, 원 패턴에 속하는 모든 백 dot 및 흑dot는 각각 1번지부터 m*n번지 사이의 번지로 표현시키게 된다. 따라서 테이블의 각행은 x번지의 dot에 해당되며 테이블의 열은 앞서 설명한 4방향의 각방향에서 끝 흑 dot의 번지를 기억하는 4개의 열과 흑 dot에 존재하는 방향의 수를 표시하는 열을 합쳐 5개의 열로 이루어진다. 물론 백 dot에 해당하는 테이블 행은 아무런 데이타도 포함치 않는다. 일단 원 패턴을 각 방향 선분(vector)으로 표현한 다음 백 dot에 해당하는 테이블 행을 삭제 시키며 또한 동일 방향에만 속하는 흑 dot의 행을 합쳐 보다 적은수의 선분으로 표시하기위한 과정을 거쳐 테이블 행을 감소시켜 나간다.

본 방식은 앞에서 한 방식의 처리단위가 dot 단위, 패턴의 선분방향과 관계없는 열(행) 단위인 것에 비해, 원 패턴을 4방향선분을 분리시킨 다음, 각 방향의 선분을 변환처리의 단위로 한 것에 주목해야 한다. 즉 dot보다 처리수준이 높은 선분의 개념을 전적으로 도입한 방식이다. 일단, 원 패턴을 vector로 표현하였으나, 테이블 크기가 매우 커지는 단점이 있다. 또한 확대에 따른 선분의 배치결과 변환 패턴상의 선분간에는 흑 dot로 매꿔야 할 부분이 백 dot로 남게 되어 품질이 열화되므로 이러한 백 dot를 채우기 위해 4방향의 수를(4방향에서 8방향으로) 증가시키지 않으면 안된다.

따라서 변환비율이 실수배이고 커짐에 따라 각 흑점에서 출발되는 방향의 수를 증가시켜야 한다. 그 결과 원 패턴을 표시하는 테이블의 행수가 증가되어 테이블 크기는 매우 커지는 단점이 있다.

그 밖에 연결시키지 않아야 할 흑 dot를 4방향 또는 8방향으로 연결시키게 되는 것을 방지키 위해 복잡한 규칙을 도입하는 결점이 있어 본 방식은 현단계로서 실용적이라고 할 수 없다.

3-7 RALTH/MACS법

원 패턴 24*24 dot의 명조체(JISC 6234 font)를 0배-2배사이의 임의 배율로 문자크기를 변환시키기 위한 연구로서 서체(font) 설계상의 규칙을(명조체 장식의 종류, 부가조건, 사선 및 종횡선분의 표현등)가능한한 일반화시켜 문자크기 변환처리에 이용한 것이 특징이다. RALTH 는 임의 배율 세선화 축소법이며 MACS는 임의 배율 확대 smoothing법이다. RALTH/MACS 는 원 패턴의 흑 dot의 위치좌표를 배율에 따라 비례적으로 사상하는 비례법과 비슷하나, 문자패턴의 변환에 따른 품질의 열화를 방지하기 위하여 세선화처리 및 윤곽선 보간처리등과 같은 처리를 부가하였다. 명조체 한자는 다른 서체의 한자와 마찬가지로 수평선분, 수직선분이 사선선분보다 많이 포함하고 있다. 그러나 명조체는 여러가지의 장식을 갖고 있다.

RALTH에서는 축소변환의 전처리과정으로서 명조체 장식의 제거(명조체를 고딕체로 변환시)와 세선화처리를 한 다음 비례법에 따라 축소변환을 행한다. 명조체 제거는 12가지의 논리mask를 이용하며 세선화 방법도 직선부분(사선제외)에 한하여 적용한다. 세선화방법은 가장 대표적인 Hilditch방식을 (분기점 distortion으로 품질이 열화된다) 택하지 않고 그림 5와 같은 논리 mask를 이용한다. 그림 5에서 0은 제거되는 명조체 장식이며 ①,②는 논리 mask에 의거 세선화를 위하여 제거되는 dot이다. MACS법은 확대용으로, 출력기기는 dot impact printer 등을 전제로 하고 있다. 비례확대법에 따라 원패턴을 변환시키는 과정에서 발생되는 보간행 또는 열(변환 패턴중원 패턴으로 부터 사상되지 않는 행 또는 열)을 인접하는 확장 열 또는 행(배율이 2배이내 이므로 보간행·열의 상하 또는 좌우에는 항상 존재한다)의 인접관계에 따라 보간하고 있다. 본 방식을 그림 6과 같은 확대 smoothing mask를 이용한다. 이는 종래 방식이 보간행(열)을 단지 인접행(열)으로 중복시키는 방식을 품질의 향상을 위해 개선한 방식이다. 그러나 변환대상이 문자이외의 도형인 경우 세선화 기법 및

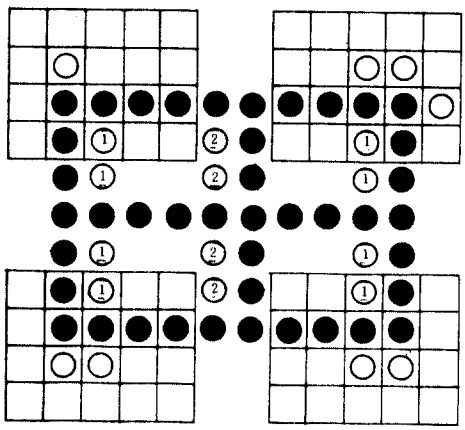

그림 5 장식제거와 세선화 예
Example of decoration elimination and line thinning

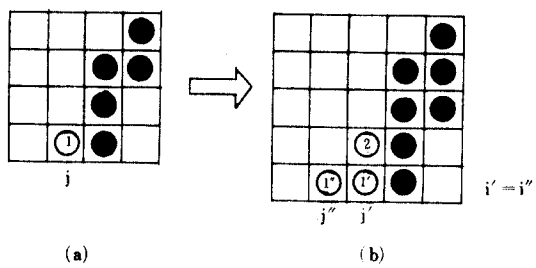

그림 6 확대 smoothing 논리 mask 적용 예
Example of adapting magnification smoothing mask.

명조체 장식제거는 축소처리에서 품질향상에 큰 도움이 되지 않으므로 자연히 본 방식은 문자처리(특히 명조체 처리)에 한정되며, 축소에 따른 사선품질의 향상을 위하여서는 개선의 여지가 있다.

4. 결 론

지금까지의 문자패턴의 크기변환에 관한 각종 연구는 주로 문자의 수가 많은 한자를 중심으로 하여 활발하게 진행되어 왔다. 지난 십여년동안 한자문화권에 속하는 나라(주로 일본)의 연구 논문을 검토해 본 결과, 문자패턴의 크기변환에 있어서 다음과 같은 연구 방향을 파악할 수 있었다.

1. 변환처리 단위가 변하고 있다. 즉 종래에는 원패턴의 dot단위로 변환처리 하는 것이 원패턴의 부분적인 특성(직선 선분, 명조체의 장식등)을 이용(추출, 제거 등)하는 경향이

많아졌다. 즉 처리단위가 dot에서 부분적으로 line단위로 바뀌고 있으며 이러한 line 단위의 처리는 기계적으로 추출된 line이 아니고 문자의 구조 분석에 따른 문자의 속성을 충분히 표현할 수 있는 line단위의 처리로 바뀌어 질 것이다.

2. 변환 대상 및 범위가 넓어지고 있다. 지금까지는 주로 한자를 중심으로 크기변환이 이루어져 왔으며 변환 비율도 인쇄용 활자의 포인터(크기)에 관련되어 이산적이며 한정된 범위에서 크기변환이 이루어졌다. 앞으로는 문자뿐만 아니라 도면, 그래프등을 포함하는 선도형의 크기 변환(회전도 포함)도 긴요할 것이며 또한, 변환범위도 연속적인 실수배의 변환처리(일부는 이미 실수배 변환하고 있음)가 일반적인 경향이다.
3. 고속화를 위한 하드웨어화; 전술의 BDM 에서 16분할 테이블법과 같이 변환처리의 부분적인 하드웨어화가 이루어지고 있으나 보다 넓은 변환대상과 범위에 있어서 지적인 변환처리를 얼마만큼 하드웨어화시킬 수 있을까 하는 것도 흥미로운 과제일 것이다.

끝으로 본 연구를 진행하는 과정에서 많은 도움을 주신 한국전자통신연구소의 강철희부장님과 김상중실장님께 감사드립니다.

李壽淵(Soo Youn LEE) 正會員

1946년 10월 24일생
1969년 2월 : 광운대학 통신공학과 졸업(공학사)
1977년 2월 : 연세대학교 대학원 전자공학과(공학석사)
1983년 3월 : 일본 교또대학 정보공학과(공학박사)
1973년 3월~현재 : 광운대학전자계산기공학과 부교수

본 논문은 85년도 한국전자통신연구소 지원으로 이루어진 "문자·도형 dot pattern의 차수변환에 관한 연구"에 관한 내용의 일부임.

참 고 문 헌

(1) 井上栄等 2人 : "漢字パターンの拡大・縮小法" 信学技報vol. 79 no. 16, pp 1~10, 1979
(2) 渡部 茂ら 3人 : "漢字パターンの拡大・縮小法の一考察"信学技報 vol. 79 no. 16, pp 11~17, 1979
(3) 森克己 : "ドット漢字パターンマトリクスの次数変換法" 信学論(D) J60-D no. 10, pp 801~808, 1977.
(4) 塩野 充 等 2人 : "漢字ドットパターン次数変換と整形の一手法" 信学論(D). J63-D. no. 7, pp 557-561, 1980.
(5) 正嶋 博 等 4人 : "二値画像の各種拡大/縮小方式の性能評価および処理速度改良方式"; 情報処理vol. 26, no. 5, pp 920~925. 1985.
(6) 間下 : "漢字ドットフオントからバクターフオントへの変換," 情報処理学会, 計算言語学研究会資料17-2, 1979.
(7) W. M Neman and R.F Sproull: "Principle of Interactive computer Graphics", McGRAW-HILL pp. 53~61, 1981.
(8) 森 克己, 中野博隆 : "直線分離法による漢字パターンの処理理," 信学論(D) J62-D no. 12, pp. 796~803, 1979. 12.
(9) 田中一男等 2人 : "文字フオントの性質に着目した任意倍率文字サイズ変換法," 信学論vol. J69-D, no. 3, pp 460~466, 1986. 3