Journal of the Korean Institute of Industrial Engineers
Vol.23, N0.4, December, 1997

완벽하고 균일하게 구성되는 암호행렬에 관한 연구*

A Study on Perfectly and Uniformly Structured Code Matrix

이성룡**

Soung Ryong Yee**

Abstract

Code matrix is the matrix of which an element and its neighbors are arranged to have a code value. The code matrix was originally designed by the author for developing a vision system but has not been theoretically studied. In this paper some theoretical properties of the code matrix are investigated. The studied characteristics of the code matrix are useful for not only understanding the matrix itself but efficiently restructuring the matrix. A number of transformation functions, which enable the matrix to have different shape, are thus developed based on the investigated properties. The transformation functions are then applied to build a perfectly and uniformly structured square code matrix, which is proven useful in an image processing example. The study in this paper is expected to serve a theoretical background for the application of the code matrix in many areas.

1. 서 론

암호행렬이란 행렬을 구성하고 있는 원소들의 특정한 조합이 암호를 형성하도록 구성원소 들을 배열한 행렬을 말한다. 그림 1은 암호행렬의 한 예를 보여주는 것으로 행렬 내 특정 위치에서의 암호는 주어진 위치에서의 원소와 그 주위의 원소값들로 구성된다. 하나의 암호는 이러한 원소들을 일정한 순서로 배열함으로써 정의되는데 예를 들어 암호를 구성하는 원소의 순서를 (주어진 위치에서의 원소, 왼쪽 원소, 위의 원소, 오른쪽 원소, 아래의 원소) 라고 한다면 그림에서 표시된 위치에서의 암호는 (1 2 3 2 3) 이 된다.

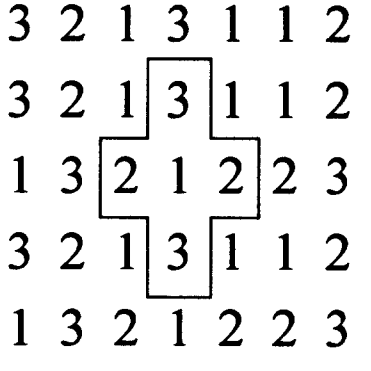


그림 1. 암호행렬과 특정 위치에서의 암호의 예

* 본 연구는 한국과학재단 핵심전문 연구비(951-1010-002-1)지원으로 수행되었음

** 한국외국어대학교 산업공학과

암호행렬은 행렬을 구성하고 있는 암호들의 구성이나 상호 연관성 등에 의해 여러 가지의 특성을 가질 수 있는데, 그 중 하나가 암호행렬 내의 어떤 암호든지 오직 한번만 존재하도록, 즉, 암호의 중복이 없도록, 구성하는 것이다. 이러한 암호행렬은 구조화된 광원(structured light)을 사용하여 삼차원의 데이터를 추출하고자 하는 비젼(vision) 시스템 개발 과정에서 상응 문제(correspondence problem)를 풀기 위하여 저자에 의해 처음으로 고안되었고, 행렬 내에서 암호의 중복이 없으므로 유일하게(uniquely) 구성되는 암호행렬이라 부른다 [1, 2, 3].

암호행렬은 발생적 기법에 의해 생성되므로 행렬 내부에 유일한 암호들이 내포되어 있다는 것 외에는 그 특성이 구조적이고 이론적으로 분석되어진 바 없다. 본 논문에서는 이러한 암호행렬이 갖는 기본적인 성격을 규명하고 이러한 행렬로부터 읽혀지는 암호의 특성을 살펴봄으로써 암호행렬의 응용 시에 필요한 이론적인 배경을 마련하고자 한다. 또한 이러한 이론적 특성을 바탕으로 암호행렬의 형태를 변화할 수 있는 변환함수를 개발하고자 한다. 개발된 변환함수를 이용하면 암호행렬 내의 원소가 균일한 형태로 배열되는 행렬을 만들 수 있는데 본 논문에서는 이러한 과정을 보여주며 결과적으로 얻어진 암호행렬을 영상처리에 응용하는 예를 또한 보여준다.

2절에서는 암호행렬을 만들기 위한 기본 형태로서의 서로 맞물린 수열에 대해 살펴보고, 이러한 수열로부터 암호행렬이 생성되는 과정을 간략히 살펴본다. 3절에서는 암호행렬이 갖는 일반적인 특성과 유일하게 구성되는 암호행렬 내에서의 암호에 관한 특성을 밝힌다. 4절에서는 암호행렬의 형태를 변화시킬 수 있는 변환 함수를 개발하고, 5절에서는 개발된 변환함수를 이용하여 행렬의 구성 원소가 균일한 형태로 배열되는 암호행렬을 구성하고 이를 영상처리에 응용하는 예를 보여준다. 끝으로 6절에서 결론을 내린다.

2. 서로 맞물린 수열과 암호행렬의 생성

2.1 서로 맞물린 수열

본 논문에서 수열이란 일정한 길이를 갖는 수의 나열을 의미하며 괄호( ) 안에 표기하기로 한다. 자연수(natural number)의 집합이 주어지고 어떤 수열을 이루는 요소(primitive)의 개수 p 가 정해질 때, 요소들의 집합을 {1, 2, 3, ..., p} 로 나타낼 수 있다. 예를 들어 p=2 의 경우 (2 2 1 2 1) 은 요소 1과 2로 만들 수 있는 수열 중 하나이다.

서로 맞물린 수열에서 '서로 맞물림' 이란 어떤 수열에서 주어진 길이의 인접한 수들이 서로 의미 있게 연관되어 있음을 뜻한다. 이러한 연관성을 나타내기 위하여 수열 내의 서로 인접하는 k 개의 요소로 이루어지는 수열을 '길이가 k 인 부분수열' 이라 정의한다.

예를 들어 수열 (2 2 1 2 1) 에서 길이가 3 인 부분수열을 순서대로 나열하면 (2 2 1), (2 1 2), (1 2 1) 이 된다. 여기에서 두 번째 부분수열 (2 1 2) 의 첫 두 요소인 2 와 1 은 첫번째 부분수열인 (2 2 1) 의 끝 두 요소인 2 와 1 이다. 마찬가지로 세 번째 부분수열 (1 2 1) 의 첫 두 요소인 1 과 2 는 두 번째 부분수열인 (2 1 2) 의 끝 두 요소인 1 과 2 이다. 즉, 이 예에서 알 수 있듯이 각 부분수열 들은 두개의 원소씩 서로 연관되어(맞물려) 있다고 말할 수 있다.

위의 예에서 수열 (2 2 1 2 1)에는 길이가 3 인 부분수열이 3개있음을 알 수 있는데, 일반적으로 전체 길이가 l 인 수열에는 길이가 k 인 부분수열이 $l-k+1$ 개 있게 된다.

주어진 수열에서는 부분수열의 길이에 따라 서로 맞물린 성질에 대한 해석이 달라질 수 있다. 예를 들어 수열 (2 2 1 2 1) 에서 길이가 3 인 부분수열은 (2 2 1), (2 1 2), (1 2 1) 로 서로 같은 부분수열을 발견할 수 없으나 길이가 2 인 부분수열을 고려하면 (2 1), (2 1), (1 2), (2 1) 로 같은 형태의 부분수열 (2 1) 이 중복됨을 알 수 있다. 따라서 원소의 개수와 부분수열의 길이에 따라 수열의 맞물림 성질은 다음과 같이 여러 가지로 구분되어 진다.

정의 1 : 주어진 p 로 구성되는 어떤 수열에서 길이가 k 인 부분수열을 고려할 때 서로 같은 부분수열을 발견할 수 없다면 유일성(uniqueness)을 갖는다고 하고 이러한 수열을 유일하게 구성되는 수열이라 부르며 $U_{(p,k)}$ 로 표기한다.

예를 들어 수열 (2 2 1 2 1)은 p=2, k=3 에 대해 유일성을 갖는 수열이다. 즉, 길이가 3인 부분수열 (2 2 1), (2 1 2), (1 2 1) 간에는 서로 중복됨이 없다. 그러나 같

은 수열을 놓고 k=2 를 고려한다면 동일한 수열 (2 1) 이 중복되므로 k=2 인 경우에는 유일하게 구성되는 수열이 아니다.

수열을 이루는 원소의 개수 p 와 부분수열의 길이 k 가 주어질 때 수열에 대한 서로 맞물린 성질 중 다른 하나로서는 완전성(completeness)을 들 수 있다.

정의 2 : 주어진 수열로부터 만들어질 수 있는 길이가 k 인 모든 부분수열의 집합이 p 개의 원소들을 사용하여 만들어 질 수 있는 길이 k 의 모든 순열(permutation)을 포함하고 있을 때 그 수열을 완전한 수열이라 부르며 $C_{(p,k)}$ 로 표기한다.

예를 들어 p=2 일 때 k=2 에 대해 완전한 수열은 길이가 2 인 부분수열 (2 2), (2 1), (1 2), (1 1) 모두를 포함하게 된다. 이러한 수열의 한 예는 (2 2 1 2 1 1) 이다. 여기서 이 수열은 완전한 수열이긴 하나 k=2 에 대해 유일한 수열은 아님을 쉽게 알 수 있다.

정의 3 : 유일성과 완전성을 모두 갖춘 수열을 완벽성(perfectness)을 갖고 있다고 정의하며, 수열을 구성하는 원소의 개수 p 와 부분수열의 길이 k 가 주어질 때 완벽한 수열은 $P_{(p,k)}$ 로 표기한다.

완벽성을 갖는 수열은 유일하게 구성되는 암호행렬을 생성하는데 기본이 된다. 원소의 개수 p 가 2 이고 k 가 2 일 때 완벽한 수열의 한 예는 (2 1 1 2 2) 이다. 이러한 완벽한 수열이 갖는 보다 상세한 이론적 특성은 [4] 에서 찾아 볼 수 있다.

2.2 암호행렬의 생성

암호행렬은 서로 맞물린 수열로부터 생성적(generative)인 기법을 사용하여 만든다. 암호행렬을 생성하는 방법은 유한상태천이(finite state transition)를 사용하는 도약연산(jump operation)으로 암호행렬의 첫째 열을 정의한 후 이 수열의 각 원소에 주어진 수만큼의 상태천이를 반복하여 새로운 열들을 생성함으로써 행렬을 구성한다. 이러한 도약연산의 개념이 다음의 그림 2 에 나타나 있다.

도약연산에 있어 암호행렬의 첫째 열과 같이 도약연산의 기본이 되는 수열을 기본수열이라 부르고 상태천이(도약)의 수를 제공하는 수열을 도약수열이라 부른다. 여기서 도약연산을 나타내는 연산기호를 ◉ 로 표기하면, 예를 들어 p=3 인 경우 수열 (1 2 3) 에서 1 만큼의 도약

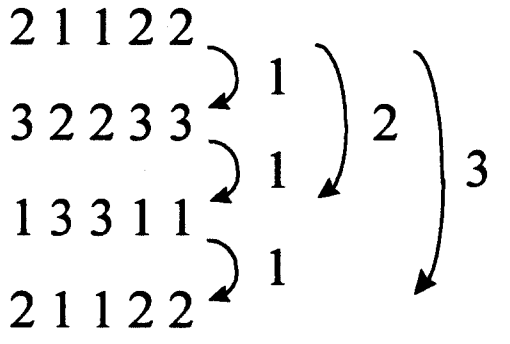


그림 2. 유한상태천이에 의한 도약연산의 개념

을 하는 것은 (1 2 3) ◉ (1) 이 되어 (2 3 1) 을 생성한다. 만약 여기에 이어서 2 만큼의 도약을 하면 (1 2 3) (=(2 3 1) ◉ (2)) 이 다시 얻어짐을 알 수 있다. 이와 같은 연속 도약의 연산을 기본수열 (1 2 3) 과 도약수열 (1 2)로 표현하면 (1 2 3) ◉ (1 2) = (1 2 3) 이 된다. 즉, 일반적으로 도약연산은 '기본수열 ◉ 도약수열' 로 나타낼 수 있다.

완벽하게 구성되는 암호행렬은 서로 맞물린 암호행렬의 대표적인 예로서 암호행렬 내에서 중복되는 암호가 발견되지 않고 또한 주어진 p 로 구성될 수 있는 모든 암호가 내재되어 있는 암호행렬을 말한다. 이러한 완벽하게 구성되는 암호행렬의 한 예는 기본수열을 $P_{(p,3)}$, 즉, 주어진 p 와 길이가 3 인 부분수열에 대해 완벽성을 지닌 서로 맞물린 수열로 하고, 도약수열을 $P_{(p,2)}$ 로 할 때 형성된다. 즉, $P_{(p,3)}$ ◉ $P_{(p,2)}$ 의 연산에 의해 하나의 완벽하게 구성되는 암호행렬이 구성되는데, 이때 주어진 p 값에 대하여 일반적인 $P_{(p,3)}$ 와 $P_{(p,2)}$ 를 발생시킬 수 있는 방법은 저자에 의해 개발된 바 있다 [1]. 이 방법에 의해 p = 3 일 때의 기본수열은 (3 3 1 3 2 1 3 1 1 2 3 1 2 2 1 2 1 1 1 3 3 2 3 2 2 2 3 3 3)을 얻게 되고 도약수열은 (3 1 2 1 1 3 2 2 3 3)을 얻게 되는데 이러한 기본수열과 도약수열을 이용하여 생성된 완벽하게 구성된 암호행렬이 그림 3 에 예시되어 있다.

3 3 1 3 2 1 3 1 1 2 3 1 2 2 1 2 1 1 1 3 3 2 3 2 2 2 3 3 3
3 3 1 3 2 1 3 1 1 2 3 1 2 2 1 2 1 1 1 3 3 2 3 2 2 2 3 3 3
1 1 2 1 3 2 1 2 2 3 1 2 3 3 2 3 2 2 2 1 1 3 1 3 3 3 1 1 1
3 3 1 3 2 1 3 1 1 2 3 1 2 2 1 2 1 1 1 3 3 2 3 2 2 2 3 3 3
1 1 2 1 3 2 1 2 2 3 1 2 3 3 2 3 2 2 2 1 1 3 1 3 3 3 1 1 1
2 2 3 2 1 3 2 3 3 1 2 3 1 1 3 1 3 3 3 2 2 1 2 1 1 1 2 2 2
2 2 3 2 1 3 2 3 3 1 2 3 1 1 3 1 3 3 3 2 2 1 2 1 1 1 2 2 2
1 1 2 1 3 2 1 2 2 3 1 2 3 3 2 3 2 2 2 1 1 3 1 3 3 3 1 1 1
3 3 1 3 2 1 3 1 1 2 3 1 2 2 1 2 1 1 1 3 3 2 3 2 2 2 3 3 3
3 3 1 3 2 1 3 1 1 2 3 1 2 2 1 2 1 1 1 3 3 2 3 2 2 2 3 3 3
3 3 1 3 2 1 3 1 1 2 3 1 2 2 1 2 1 1 1 3 3 2 3 2 2 2 3 3 3

그림 3. p=3일 때 완벽하게 구성되는 암호행렬

3. 암호행렬의 특성

암호행렬을 각 열(column)의 관점에서 보면 각 열은 어떤 원소의 값에 도약수열을 적용시켜 생성됨을 안다. 예를 들어 (1) ◉ (3 1 2) 는 $(1\ 1\ 2\ 1)^T$ 이란 열을 생성한다. 이러한 도약연산은 다음의 정리에서 나타나는 것과 같은 특성을 갖는다.

정리 1 : 도약연산 $(n) \circledcirc U_{(p,\ k)}$ 는 또 다른 하나의 $U_{(p,\ k+1)}{}^T$ 를 생성한다.

증명 : 도약수열 $U_{(p,\ k)} = (1, 2, ..., l)$ 라하고, $U_{(p,\ k)}$ 에서 두개의 서로 다른 부분수열 $(\delta_i, \delta_{i+1}, ..., \delta_{i+k-1})$ 과 $(\delta_j, \delta_{j+1}, ..., \delta_{j+k-1})$, $i \neq j$ 를 정의하자. 이때 부분수열의 길이가 k 이면 도약연산의 결과는 $k+1$ 인 수열이 됨을 안다. $(\delta_i, \delta_{i+1}, ..., \delta_{i+k-1})$ 을 사용한 도약연산에서 길이가 $k+1$ 인 부분수열 $(x_1, x_2, ..., x_{k+1})^T$ 가 얻어지고 $x_2 = x_1 \circledcirc (\delta_i)$ 라 한다면, 이 부분수열이 유일하다는 것을 보이는 것은 $U_{(p,\ k)}$ 를 적용한 전체 연산 결과에서 $(x_1, x_2, ..., x_{k+1})^T$ 와 같은 부분수열이 다른 곳에서는 존재하지 않음을 보이는 것과 같다.

만약 $(x_1, x_2, ..., x_{k+1})^T$ 와 같은 부분수열이 다른 곳에 존재하고 이것을 $y_2 = y_1 \circledcirc (\delta_j)$ 인 $(y_1, y_2, ..., y_{k+1})^T$ 라 하면 $x_1 = y_1$, $x_2 = y_2$, ..., $x_{k+1} = y_{k+1}$ 이 된다. 따라서 $x_1 \circledcirc (\delta_i) = x_2 = y_2 = y_1 \circledcirc (\delta_j) = x_1 \circledcirc (\delta_j)$ 가 되므로 $|\delta_i - \delta_j| = mp$, $m \in N$ 가 된다. 여기서 $\delta_i, \delta_j \leq p$ 가 됨을 알고 있으므로 $0 \leq |\delta_i / p - \delta_j / p| = m < 1$ 이 되어 이러한 관계식을 만족하는 가능한 m 값은 0 뿐이다. 따라서 $\delta_i = \delta_j$가 되고 이와 같은 과정을 모든 x 와 y 에 대해서 수행한다면 $(\delta_i, \delta_{i+1}, ..., \delta_{i+k-1}) = (\delta_j, \delta_{j+1}, ..., \delta_{j+k-1})$ 라는 결론에 도달하게 되어 $U_{(p,\ k)}$ 가 유일성을 갖는다는 것과 모순된다.■

정리 2 : $n \in \{1, 2, ..., p\}$ 인 모든 n 에 대하여 도약연산 $(n) \circledcirc P_{(p,\ k)}$ 로부터 얻어지는 각 수열은 길이가 $k+1$ 인 부분수열에 대해 유일성은 보존하나 완전하지는 않다. 그러나 서로 상호 보완적(complementary)으로 이들 모두는 길이가 $k+1$ 인 부분수열의 완전한 집합을 형성한다.

정리 2를 증명하기에 앞서 정리 2가 의미하는 바를 예를 통해 살펴보면 다음과 같다.

예 : $p=3$ 이라 하자. (1) ◉ $P_{(3,\ 2)}$ 는 (1) ◉ (3 1 2 1 1 3 2 2 3 3) = $(1\ 1\ 2\ 1\ 2\ 3\ 3\ 2\ 1\ 1\ 1)^T$ 가 되고, (2) ◉ $P_{(3,\ 2)}$ 는 $(2\ 2\ 3\ 2\ 3\ 1\ 1\ 3\ 2\ 2\ 2)^T$ 가 되고, (1) ◉

$P_{(3,\ 2)}$ 는 (1) ◉ (3 1 2 1 1 3 2 2 3 3) = (3 3 1 3 1 2 2 1 3 3 3)^T 가 된다. 이때 얻어진 세 개의 수열들은 길이가 3 인 부분수열에 대해 각각 유일성을 갖고 있다. 이 세 개의 수열로부터 길이가 3 인 부분수열 들을 구해보면 서로 중복됨이 없이 {1, 2, 3}으로부터 형성될 수 있는 모든 순열을 포함하는 완전한 집합이 됨을 안다.

증명 : 완벽성을 갖는 수열은 정의에 의해 유일성을 갖고 있음을 알고 정리 1에 의해 이러한 수열을 도약수열로 하여 형성되는 수열 또한 유일성을 갖게 됨을 안다. 여기서 얻어진 수열이 완전하다면 얻어진 수열은 $P_{(p,\ k+1)}$ 이 되므로 수열의 길이가 $p^{k+1}+k$ 임을 쉽게 알 수 있는데, 도약연산 (n) ◉ $P_{(p,\ k)}$ 로부터 얻어진 수열의 길이는 $(p^k+k-1)+1 = p^k+k$ 가 되어 얻어진 수열은 완전하지 않음을 안다.

이제 각 n 과 여기에 도약연산을 적용시켜 얻어진 각각의 수열에는 길이가 $k+1$ 인 부분수열이 몇 개씩 존재하는지 살펴본다. 만약 도약연산에 의해 완벽한 수열이 생성되었다면 $P_{(p,\ k+1)}$ 가 되고, 그 안에서 길이가 $k+1$ 인 수열은 $(p^{k+1}+k)-(k+1)+1$ 개가 발생한다. 그러나 각각의 생성된 수열에는 $(p^k+k)-(k+1)+1$ 개의 부분수열이 있다. 여기서 이들의 비(比) $[(p^{k+1}+k)-(k+1)+1] / [(p^k+k)-(k+1)+1]$ 를 구해보면 p 가 되고 이는 곧 $|P|$ 임을 알 수 있다. 또한 각각의 발생된 수열은 유일성을 갖고 각기 서로 다른 원소로부터 도약연산에 의해 생겨났으므로 중복된 부분수열이 없음을 안다. 따라서 도약연산 (n) ◉ $P_{(p,\ k)}$ 로부터 얻어지는 수열들은 길이가 $k+1$ 인 부분수열의 완전한 집합을 형성한다.■

이제까지 도약연산에 의해 특징 지워지는 암호행렬의 특성에 대해 살펴보았는데, 이를 기반으로 암호행렬 내의 암호의 특성을 알아볼 수 있다. 다음은 암호들의 서로 맞물린 성질과 이제까지 살펴본 행렬 자체의 성질을 가장 잘 나타낼 수 있는 완벽하게 구성되는 암호행렬에 대한 암호의 구성 특성을 정리한 것이다.

정의 4 : $m\times n$ 의 완벽하게 구성되는 암호행렬의 i 행과 j 열 위치에서의 암호는 $(x_{i\ j}, x_{i\ j-1}, x_{i-1\ j}, x_{i\ j+1}, x_{i+1\ j})$, $i \in \{1,2, ..., m-1\}$, $j \in \{1,2, ..., n-1\}$ 로 정의하고 W_{ij} 로 표기한다. 또한 $W_{ij}^{\ v}$ 는 암호의 구성원소 중 세로 방향의 수열로 $(x_{i-1\ j}, x_{i\ j}, x_{i+1\ j})$를 $W_{ij}^{\ h}$ 는 가로 방향의 수열로 $(x_{i\ j-1}, x_{i\ j}, x_{i\ j+1})$을 나타낸다.

위의 정의와 정리 1 로부터 다음의 따름 정리를 이끌어 낼 수 있다.

따름정리 1−1 : 완벽하게 구성되는 암호행렬 내의 어떠한 한 암호를 구성하는 세로 방향의 수열 $W_{i,j}^{\ v}$ 은 그 수열을 포함하는 j 열에서 유일하게 존재한다.

gothic : 완벽하게 구성되는 암호행렬은 $P_{(p,\ 3)}$ ◉ $P_{(p,\ 2)}$ 의 결과로부터 얻어지는 행렬로서 그림 1 에서 보듯이 한 암호를 가로 방향의 세 개의 수와 세로 방향의 세 개의 수로 볼 수 있다. 이때 이들 각각은 행과 열의 부분수열이므로 정리 1 에 의해 암호를 구성하는 세로 방향의 수열은 그 수열을 포함하는 열에서 유일하게 존재함을 안다.■

정리 3 : 완벽하게 구성되는 암호행렬에서 $W_{i,j}^{\ v}$에 도약연산을 시켜 얻어질 수 있는 수열은 오직 그 수열이 나타난 행, 즉, i 행에서만 나타난다.

증명 : $W_{i,j}^{\ v} = (x_{i-1\ j}, x_{ij}, x_{i+1\ j})^T$ 에 도약연산을 적용하여 만들어지는 수열을 $(x_{k-1\ l}, x_{kl}, x_{k+1\ l})^T$ 이라 할 때 $i = k$ 임을 보이면 된다.

$W_{i,j}^{\ v}$ 에 도약수열 (n) 을 적용시켜 도약연산을 행하면 $x_{k-1\ l} = x_{i-1\ j}$ ◉ (n), $x_{kl} = x_{ij}$ ◉ (n), $x_{k+1\ l} = x_{i+1\ j}$ ◉ (n) 이다. 이 때 x_{kl} 은 $x_{k-1\ l}$ 에 (δ_{k-1}) 의 도약을 적용시켜 얻어진 것이라는 것을 상기하면, $x_{kl} = x_{k-1\ l}$ ◉ $(\delta_{k-1}) = ((x_{i-1\ j})$ ◉ $(n))$ ◉ $(\delta_{k-1}) = (x_{i-1\ j})$ ◉ $(n+\delta_{k-1})$ 이며, 또한 같은 논리로 $x_{kl} = x_{ij}$ ◉ $(n) = ((x_{i-1\ j})$ ◉ $(\delta_{i-1}))$ ◉ $(n) = (x_{i-1\ j})$ ◉ $(n+\delta_{i-1})$ 이다. 따라서 $|(n+\delta_{k-1}) - (n+\delta_{i-1})| = mp$, $m \in N$ 이 됨을 알 수 있는데 $0 \le |\delta_{k-1} / p - \delta_{i-1} / p| = m < 1$ 이므로 m 은 0 이어야 한다. 따라서 $\delta_{k-1} = \delta_{i-1}$ 이고 $\delta_k = \delta_i$ 가 된다. 만약 $i \neq k$ 이라면 이것은 유일하게 구성되는 암호행렬을 구성하는데 사용된 도약수열이 유일하지 않다는 것이므로 모순된다.■

정리 3은 유일하게 구성되는 암호행렬에서 특정 암호를 쉽고 빠르게 찾는데 이용될 수 있는 성질을 제공한다. 정리 3의 논리를 행에 대해서도 적용한다면, 유일하게 구성되는 암호행렬은 암호행렬의 생성에 사용되는 기본수열이 완벽성을 갖고 이 기본수열로부터 도약연산이 적용되어 나머지 열들이 생성되므로 임의의 암호에 대해 행의 위치 j 는 그 암호를 구성하는 가로 방향의 수열 $W_{ij}^{\ h}$ 로부터 찾아낼 수 있음을 안다. 이러한 논의로부터 다음

의 방법이 제시된다.

gothic : 완벽하게 구성되는 암호행렬에서 임의의 암호 $W_{i,j}$ 의 위치 (i, j)는 행렬 내에서 유일하게 결정되며, 행의 값 i 는 암호의 $W_{i,j}^{r}$ 로부터 도약연산을 시켜 얻을 수 있는 수열을 암호행렬의 첫 열로부터 찾고, 열의 값 j 는 $W_{i,j}^{k}$ 로부터 도약연산을 시켜 얻을 수 있는 수열을 암호행렬의 첫 행으로부터 찾는다.

4. 암호행렬의 변환

암호행렬은 기본수열로부터 유한상태변이를 순차적으로 행함으로써 얻어지므로 암호행렬의 형태는 기본수열과 상태변이의 횟수를 결정하는 도약수열에 의해 결정된다. 생성적 기법[1]에 의해 형성된 암호행렬은 기본수열과 도약수열에서 사용하는 원소의 개수가 p라 할 때, 그림 3에서 보듯이 $(p^2+2)\times(p^3+2)$ 의 직사각형의 형태를 갖게 된다.

암호행렬을 실제로 응용하고자 하는 경우 이러한 암호행렬의 일정한 형태가 제한 요인으로 작용할 수 있다. 예를 들어 암호행렬을 삼차원 물체 측정을 위한 비젼시스템에서 응용하는 경우 암호행렬을 광원으로써 사용하게 되는데 [3], 직사각형의 광원보다는 정방형의 광원이 더욱 유용함을 알 수 있다. 따라서 생성적 기법에 의해 만들어진 암호행렬을 인위적인 조작에 의해 형태 변환을 시킬 수 있는 방법의 연구가 필요하게 되며 형태변환을 유도할 수 있는 변환 함수의 개발이 하나의 접근방법이 된다.

기본적으로 암호행렬은 서로 맞물린 수열로부터 생성되므로 수열 형태의 변환이 곧 암호행렬의 형태 변환으로 이어질 수 있다는 데에 착안하여 다음의 여러 가지 형태의 변환함수를 개발하였다.

4.1 도약에 의한 변환

전술한 바와 같이 도약에 의한 변환은 유한상태 변이를 시키는 함수로 정의할 수 있다. 즉, $\mathcal{T}_{(n)}(S)$ 를 수열 S 를 n 만큼 도약시키는 함수라고 정의하면 $\mathcal{T}_{(n)}(S) = S \circledcirc (n)$ 이 된다. 예를 들어 완벽한 수열 $P_{(3,\ 2)}$ = (3 1 2 1 1 3 2 2 3 3) 에 (1)의 도약을 적용하는 변환은 $\mathcal{T}_{(1)}(P_{(3,\ 2)})$ 로 나타낼 수 있으며 (3 1 2 1 1 3 2 2 3 3) ◉ (1)가 되어 (1 2 3 2 2 1 3 3 1 1)의 수열로 변환된다.

참고적으로 위의 예에서 보듯이 완벽한 수열을 도약변환시키면 또 다른 완벽한 수열이 얻어짐을 알 수 있다. 이것은 언제나 성립하는 성질로서 부분수열이 갖는 성질은 유한상태 변이에 의해 바뀌지 않고 따라서 변환된 수열이 유일성을 보전하기 때문이다. 또한 도약에 의한 변환을 이용하여 완벽한 수열 $P_{(p,\ k)}$ 로부터 $p-1$ 개의 서로 다른 완벽한 수열을 생성할 수 있음을 쉽게 알 수 있다.

암호행렬에 적용되는 도약에 의한 변환은 곧 암호행렬을 구성하고 있는 원소 각각에 대해 유한상태 변이를 시키는 것이다. 예를 들어 완벽하게 구성되는 암호행렬 $P_{(p,\ 3)}$ ◉ $P_{(p,\ 2)}$ 에 (n) 만큼의 도약변환은 $\mathcal{T}_{(n)}(P_{(p,\ 3)} \circledcirc P_{(p,\ 2)})$ 로 나타낼 수 있으며 변환 결과로 얻어지는 암호행렬은 역시 완벽하게 구성되며, 같은 형태의 암호행렬이 $\mathcal{T}_{(n)}(P_{(p,3)})$ ◉ $P_{(p,\ 2)}$ 로부터도 얻어짐을 알 수 있다.

4.2 회전에 의한 변환

수열의 형태를 2차원 평면에서 한 줄로 쓰여지는 고정된 형태로만 보지 않을 때 흥미로운 변환함수를 유도할 수 있다. 그림 4에서 보듯이 수열이 쓰여있는 종이를 잘라 구부려서 수열의 앞과 끝을 겹친 상태로 연결시키고 반지 모양의 종이를 적당한 위치에서 자르고 다시 펴면 전혀 새로운 형태의 수열이 얻어지게 되는데 이러한 변환을 회전에 의한 변환이라 부르기로 한다.

회전에 의한 변환은 $\mathcal{R}_{(r)}(S)$로 표기하며 S 는 대상 수열을, r 은 회전의 정도를 나타낸다. 즉, $\mathcal{R}_{(r)}(S)$ 는 주어진 수열 S 의 $r+1$ 번째 원소가 수열의 맨 처음으로 오게 하는 회전을 유도하게 된다.

여기서 완벽한 수열은 회전에 의한 변환에 의해서도 또 다른 완벽한 수열을 생성하게 됨이 흥미롭다. 완벽한 수열 $P_{(p,\ k)}$ 의 양쪽 끝 $k-1$ 개의 원소는 언제나 서로 같게 되는데 [4], 이들을 서로 겹치게 하여 반지 모양을 만든 후 반지를 회전시켜 자르고 펴면 새로운 일직선의 수열이 형성된다. 이 수열의 앞 $k-1$ 개의 원소와 같은 부분수열을 새로이 만들어진 수열의 맨 뒤에 덧붙이기만 하면 처음과는 전혀 다른 새로운 완벽한 수열이 구성된다. 즉, 완벽한 수열 $P_{(p,\ k)}$ 는 회전에 의한 변환에 의해 p^k 개의 새로운 완벽한 수열을 생성할 수 있다.

예를 들어 그림 4는 $P_{(3,\ 2)}$ 의 회전에 의한 변환을 보여

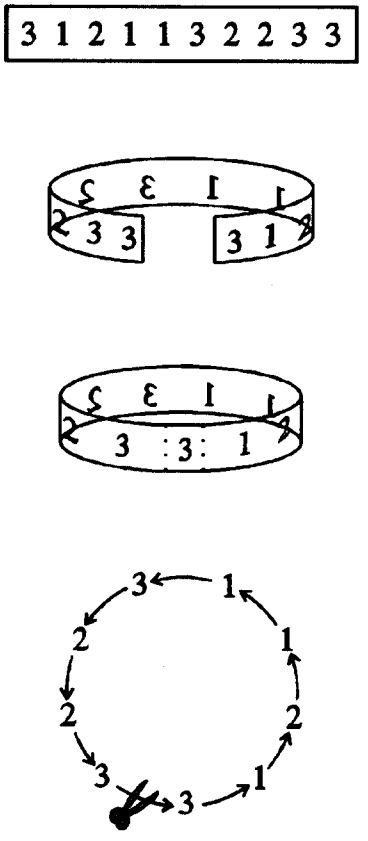


그림 4. 회전 변환의 개념

주고 있다. 그림에서 가위로 자르는 부분은 r = 0 의 위치로서 $\mathfrak{R}_{(0)}(P_{(3,\ 2)})$의 변환을 나타내고 있다. 즉, 그림의 맨 밑에 형성된 수열의 맨 끝에 수열의 처음 원소를 붙이면 원래의 완벽한 수열이 됨을 쉽게 알 수 있다. 그림의 수열에 대한 다른 회전 변환의 예로는 $\mathfrak{R}_{(1)}(P_{(3,\ 2)})$=(1 2 1 1 3 2 2 3 3), $\mathfrak{R}_{(2)}(P_{(3,\ 2)})$=(2 1 1 3 2 2 3 3 1), $\mathfrak{R}_{(3)}(P_{(3,\ 2)})$= (1 1 3 2 2 3 3 1 2) 등을 들 수 있다.

임의의 암호행렬 M 에 대한 회전 변환은 수평방향에 대한 회전 $\mathfrak{R}^{h}_{(r)}(S)$ 와 수직방향에 대한 회전 $\mathfrak{R}^{v}_{(r)}(S)$ 로 구분할 수 있다. 각 방향에 대한 회전의 예를 그림 5에 나타내었는데 그림의 암호행렬은 원소의 개수가 2인 완벽하게 구성되는 암호행렬이다.

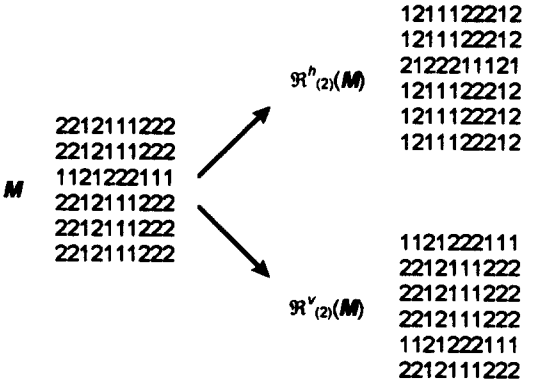


그림 5. 암호행렬의 회전에 의한 변환

암호행렬은 행렬의 첫 번째 열로부터 도약변환에 의해 생성되므로 임의의 암호행렬 전체에 대한 수평방향의 회전은 행렬의 첫 번째 열을 우선 회전 변환시킨 후에 도약 변환을 적용한 것과 같다. 예를 들어 완벽하게 구성되는 암호행렬 $P_{(p,\ 3)} \circledcirc P_{(p,\ 2)}$의 수평방향 회전 변환인 $\mathfrak{R}^{h}_{(r)}(P_{(p,\ 3)} \circledcirc P_{(p,\ 2)})$는 $\mathfrak{R}^{h}_{(r)}(P_{(p,\ 3)}) \circledcirc P_{(p,\ 2)}$ 와 같은 결과를 갖는다. 여기서 흥미로운 것은 이와 같이 완벽하게 구성되는 암호행렬을 회전변환 시켰을 때 얻어지는 새로운 암호행렬은 여전히 완벽하게 구성된다는 사실이다. 변환 후에도 이러한 특성이 보존되는 것은 중요한 특성으로 암호행렬의 형태 변환에 의미를 부여하게 된다.

4.3 자름함수와 붙임연산

암호행렬의 형태 변환을 위해 도약변환과 회전변환 이외에 자름함수와 붙임연산을 정의한다. 자름함수는 $\aleph_{(q)}(S)$ 로 나타내고 수열 S 의 끝 부분 q개의 원소를 잘라내는 변환을 한다. 예를 들어 S 가 (3 1 2 1 1 3)일 때 $\aleph_{(2)}(S)$ 는 (3 1 2 1)의 결과를 갖는다.

회전변환함수와 유사하게 행렬 M 에 대해서는 자름함수가 수평, 수직 방향으로 정의되고 수평 방향의 자름 $\aleph^{h}_{(q)}(M)$ 은 행렬의 오른쪽 끝으로부터 q개의 열(column)을 삭제하는 것으로 수직 방향의 자름 $\aleph^{v}_{(q)}(M)$ 은 행렬의 아래쪽으로부터 q개의 행(row)을 삭제하는 것으로 정의한다.

붙임연산은 두 개의 서로 다른 수열이나 행렬을 붙이

는 연산으로, ⊕의 기호로 나타내기로 한다. 예를 들어 (1 2 1)⊕(2 2 1)은 (1 2 1 2 2 1)이 된다. 행렬에서는 자름함수와 마찬가지로 수평과 수직 방향의 붙임연산으로 나누어 질 수 있는데 특별히 구분하여 정의하지 않고 자름함수의 적용 직후에 또는 회전변환함수의 적용 직후에 사용하는 경우 자름함수나 회전변환함수의 방향과 같은 것으로 가정한다. 이러한 연산의 예는 다음의 그림 6을 통해서 찾아볼 수 있다.

첫 번째 행 즉, 기본수열인 $P_{(2,\ 3)}$를 구성하고 있는 모든 부분수열을 포함하게 하기 위해서이다.

이 과정을 다시 설명하면 (2 2 1 2)⊕(2 2)=(2 2 1 2 2 2) 와 (1 1 1 2)⊕(1 1)=(1 1 1 2 1 1) 이 되는데, 이러한 과정을 행렬 M의 모든 행에 대해 수행하면 그림 6에서 보듯이 M^1 과 M^2 두 개의 부분행렬이 형성된다. 이 두 개의 부분행렬을 단순히 서로 붙인다면 12×6의 행렬이 구성되는데 이렇게 구성되는 행렬은 서로 중복되는 암

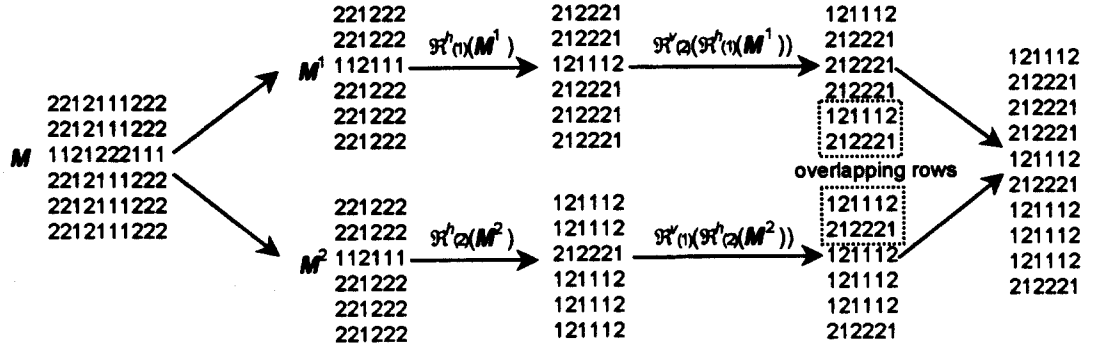


그림 6. 암호행렬의 형태 변환 예

4.4 형태 변환의 예

그림 6에서의 행렬 M은 원소의 개수가 2 일 때 완벽한 수열인 $P_{(2,\ 3)}$=(2 2 1 2 1 1 1 2 2 2)를 기본수열로 하고 $P_{(2,\ 2)}$=(2 1 1 2 2)를 도약수열로 하여 생성된 완벽하게 구성되는 암호행렬이다. 그림에서 알 수 있듯이 행렬 M의 크기는 6×10 이다. 이 암호행렬이 갖고 있는 완벽하게 구성되는 특성은 보전하면서 다른 크기의 암호행렬로 형태 변환을 하고자 한다면 아마도 가장 간단하게 시도할 수 있는 방법은 단순히 전치행렬(transposition)로 바꾸는 것이고 이 변환에 의해 10×6 형태의 행렬을 얻게 된다.

이제 위에서 정의된 함수들을 이용하여 형태 변환을 하고자 한다. 우선 행렬 M의 첫 번째 행에 자름함수를 적용하여 새로운 수열을 만든다. 즉, $\aleph_{(2)}(P_{(2,\ 3)})$=(2 2 1 2 1 1 1 2)를 얻고 이 수열을 반으로 잘라 (2 2 1 2) 와 (1 1 1 2)의 새로운 두 개의 수열을 구성한다. 이렇게 새로 만들어진 수열들 각각에 대해 처음 두 개의 원소를 뒤에 붙이는 연산을 적용한다. 이는 두 개의 수열이 행렬 M의 호들을 갖게 되어 완벽하게 구성되지 않음을 쉽게 알 수 있다. 따라서 그림에서와 같이 두 부분수열이 공통부분을 갖도록 각각의 부분행렬을 수평 및 수직 방향으로 회전변환시킨 후에 공통부분을 중첩시키면 전혀 새로운 형태의 완벽하게 구성되는 암호행렬이 구성된다. 이러한 중첩은 전 절에서 정의된 자름함수와 붙임연산으로 표현할 수 있는데 그림에서의 과정은 $\Re^{v}_{(2)}(\Re^{v}_{(2)}(\Re^{h}_{(1)}(M^1)))\oplus\Re^{v}_{(1)}(\Re^{h}_{(2)}(M^2))$ 로 나타낼 수 있다.

우연한 경우이긴 하지만 이 예에서 복잡한 변환을 통해 구해진 암호행렬은 처음의 M을 회전 변환을 시킨 후에 단순히 전치 시킴으로써도 얻어질 수 있다. 즉, 최종에 얻어진 행렬은 $[\Re^{v}_{(1)}(\Re^{h}_{(6)}(M))]^T$ 로부터도 얻어질 수 있음이 흥미롭다.

5. 균일한 암호행렬의 구성 및 응용

4절에서 개발된 변환 함수에 의해 완벽하게 구성되는 암호행렬로부터 균일한(uniform) 특성을 갖는 암호행렬의

구축을 시도해 볼 수 있다. 균일한 암호행렬이란 행렬을 구성하고 있는 각 원소들이 행렬내에서 균일하게 나타나고, 나타나는 형태에 있어서도 일정한 패턴이 없음을 의미한다.그림 7은 균일성을 갖고 있지 않은 행렬들의 예를 보여주고 있다. 그림에서 (a)에 나타난 행렬은 정방형의 암호행렬 내에서 원소별로 동일한 개수가 사용되지 않았으므로 균일한 행렬이라 볼 수 없으며, (b)에 나타난 행렬은 원소별로 동일하게 사용은 되었으나 그 분포가 일정한 패턴을 갖고 있어 역시 균일한 행렬로 볼 수 없다.

생성적 기법[1]을 통해 형성되는 암호행렬의 균일성은 첫째, 암호행렬을 구성하는 기본수열과 도약수열의 구성

1	1	1	3	1	1	3	1	1
1	1	1	3	1	1	3	1	1
1	1	2	1	1	2	1	2	2
3	3	1	3	1	1	3	1	1
1	1	2	1	3	2	1	2	2
1	1	3	2	1	3	2	1	1
1	1	3	2	1	3	2	1	1
1	1	2	1	1	2	1	2	2
3	3	1	3	1	1	3	1	1

(a) 각 원소 사용의 개수가 균일하지 않은 행렬

1	1	1	2	2	2	3	3	3
1	1	1	2	2	2	3	3	3
1	1	1	2	2	2	3	3	3
2	2	2	1	1	1	3	3	3
2	2	2	1	1	1	3	3	3
2	2	2	1	1	1	3	3	3
3	3	3	2	2	2	1	1	1
3	3	3	2	2	2	1	1	1
3	3	3	2	2	2	1	1	1

(b) 원소분포가 특정 패턴을 따르는 행렬

그림 7. 균일하지 않은 암호행렬의 예

원소들에 대한 균일성을 1차원 선상에서 균일하게 분포하는지를 검토하고 둘째, 암호행렬의 각 구성원소들이 2차원 평면에서 균일하게 분포하는 지를 검토함으로써 알 수 있다. 우선 수열에 대한 균일성은 런 테스트에 의해 일정 패턴이 없음을 보이고, 이러한 수열이 생성적으로 형성하는 행렬 또한 균일할 것이라는 가정 하에, 행렬에 대해서는 카이제곱 적합도 검정, 그리고 무게중심의 계산에 의해 균일성을 밝힐 수 있다. 이러한 과정에 대한 상세한 내용은 [6]에서 찾을 수 있다.

본 논문에서는 생성적으로 만들어지는 직사각형 모양의 암호행렬을 변환함수에 의해 정방형으로 구성하고 이렇게 구성된 암호행렬에 대한 균일성에 대해 검토하기로 한다.

5.1 완벽하고 정방형인 암호행렬의 형성

완벽하게 구성되는 행렬은 유일하게 구성되는 행렬일 뿐만 아니라 행렬 내에 원소들로부터 만들어 질 수 있는 가능한 모든 암호가 포함되어 있음은 전술한 바와 같다. 따라서 완벽하며 또한 균일하게 구성되는 정방형(square)의 암호행렬을 형성하고자 하는데 행렬의 형태는 암호행렬에 사용되는 원소의 개수에 민감함에 주의할 필요가 있다. 예를 들어 p=3 인 경우라면 완벽한 행렬의 형태는 3×81, 81×3, 9×27, 27×9 만을 취할 수 있고 따라서 이로부터는 완벽하게 구성되는 정방형의 행렬을 유도할 수는 없다. 한편 원소의 개수가 네 개인 p=4 인 경우에는 암호의 개수가 4^5(=1024=32×32) 이 되므로 (32+2)×(32+2) 형태의 정방형 암호행렬을 구성할 수 있다. 일반적으로 정방형의 완벽한 수열은 $p=q^2$, $q\in$N인 경우에만 형성될 수 있음을 안다. 다음은 p=4 인 경우 완벽한 정방형의 암호행렬을 형성하는 한 과정을 보여주고 있다.

step1 :

p=4 일 때의 완벽한 수열 $P_{(4,\ 2)}$=(4 1 3 1 2 1 1 4 2 3 2 2 4 3 3 4 4)를 생성적 기법에 의해 형성한다.

step2 :

p 의 값 각각에 대해 단일 원소를 갖는 수열을 기본수열로 하고, $P_{(4,\ 2)}$를 도약수열로 하는 도약연산을 통하여 상호보완적(complementary)인 수열들을 생성한다. 즉,

S_1=(1)◉$P_{(4,\ 2)}$=(1 1 2 1 2 4 1 2 2 4 3 1 3 3 2 1 1 1),

S_2=(2)◉$P_{(4,\ 2)}$=(2 2 3 2 3 1 2 3 3 1 4 2 4 4 3 2 2 2),

S_3=(3)◉$P_{(4,\ 2)}$=(3 3 4 3 4 2 3 4 4 2 1 3 1 1 4 3 3 3),
S_4=(4)◉$P_{(4,\ 2)}$=(4 4 1 4 1 3 4 1 1 3 2 4 2 2 1 4 4 4)
를 생성한다.

step3 :

정방형의 행렬을 구성하게 될 부분행렬(sub-matrix)들을 위에서 구한 상호보완적 수열들과 변환함수를 이용하여 형성한다. 즉, 부분행렬을 생성할 기본수열 중 하나인 SS_1 은 $\aleph_{(2)}(S_2)\oplus\aleph_{(2)}(\mathfrak{R}_{(12)}(S_4))\oplus(2\ 2)$ 로부터 그리고 SS_2 는 $\aleph_{(2)}(S_1)\oplus\aleph_{(2)}(\mathfrak{R}_{(12)}(S_3))\oplus(1\ 1)$ 로부터 만들고, 이들 기본수열과 도약수열 $P_{(4,\ 2)}$를 사용하여 부분행렬 $M^1=SS_1$ ◉ $P_{(4,\ 2)}$ 와 $M^2=SS_2$ ◉ $P_{(4,\ 2)}$ 를 유도한다.

step4 :

M^1 은 정방형 행렬의 상위 반쪽을 구성하는 부분행렬이 된다. 정방형 행렬의 하위 반쪽은 $\mathfrak{R}^{v}_{(7)}(M^2)$ 로 M^2 를 변환하여 M^1 의 마지막 두 개의 열과 변환된 부분행렬의 처음 두 개의 열을 같도록 한다. 이제 이 두 개의 부분행

2	2	3	2	3	1	2	3	3	1	4	2	4	4	3	2	2	2	1	4	4	4	1	4	1	3	4	1	1	3	2	4	2	2
2	2	3	2	3	1	2	3	3	1	4	2	4	4	3	2	2	2	1	4	4	4	1	4	1	3	4	1	1	3	2	4	2	2
3	3	4	3	4	2	3	4	4	2	1	3	1	1	4	3	3	3	2	1	1	1	2	1	2	4	1	2	2	4	3	1	3	3
2	2	3	2	3	1	2	3	3	1	4	2	4	4	3	2	2	2	1	4	4	4	1	4	1	3	4	1	1	3	2	4	2	2
3	3	4	3	4	2	3	4	4	2	1	3	1	1	4	3	3	3	2	1	1	1	2	1	2	4	1	2	2	4	3	1	3	3
1	1	2	1	2	4	1	2	2	4	3	1	3	3	2	1	1	1	4	3	3	3	4	3	4	2	3	4	4	2	1	3	1	1
2	2	3	2	3	1	2	3	3	1	4	2	4	4	3	2	2	2	1	4	4	4	1	4	1	3	4	1	1	3	2	4	2	2
3	3	4	3	4	2	3	4	4	2	1	3	1	1	4	3	3	3	2	1	1	1	2	1	2	4	1	2	2	4	3	1	3	3
3	3	4	3	4	2	3	4	4	2	1	3	1	1	4	3	3	3	2	1	1	1	2	1	2	4	1	2	2	4	3	1	3	3
1	1	2	1	2	4	1	2	2	4	3	1	3	3	2	1	1	1	4	3	3	3	4	3	4	2	3	4	4	2	1	3	1	1
4	4	1	4	1	3	4	1	1	3	2	4	2	2	1	4	4	4	3	2	2	2	3	2	3	1	2	3	3	1	4	2	4	4
2	2	3	2	3	1	2	3	3	1	4	2	4	4	3	2	2	2	1	4	4	4	1	4	1	3	4	1	1	3	2	4	2	2
4	4	1	4	1	3	4	1	1	3	2	4	2	2	1	4	4	4	3	2	2	2	3	2	3	1	2	3	3	1	4	2	4	4
4	4	1	4	1	3	4	1	1	3	2	4	2	2	1	4	4	4	3	2	2	2	3	2	3	1	2	3	3	1	4	2	4	4
3	3	4	3	4	2	3	4	4	2	1	3	1	1	4	3	3	3	2	1	1	1	2	1	2	4	1	2	2	4	3	1	3	3
2	2	3	2	3	1	2	3	3	1	4	2	4	4	3	2	2	2	1	4	4	4	1	4	1	3	4	1	1	3	2	4	2	2
2	2	3	2	3	1	2	3	3	1	4	2	4	4	3	2	2	2	1	4	4	4	1	4	1	3	4	1	1	3	2	4	2	2
2	2	3	2	3	1	2	3	3	1	4	2	4	4	3	2	2	2	1	4	4	4	1	4	1	3	4	1	1	3	2	4	2	2
4	4	1	4	1	3	4	1	1	3	2	4	2	2	1	4	4	4	3	2	2	2	3	2	3	1	2	3	3	1	4	2	4	4
3	3	4	3	4	2	3	4	4	2	1	3	1	1	4	3	3	3	2	1	1	1	2	1	2	4	1	2	2	4	3	1	3	3
1	1	2	1	2	4	1	2	2	4	3	1	3	3	2	1	1	1	4	3	3	3	4	3	4	2	3	4	4	2	1	3	1	1
3	3	4	3	4	2	3	4	4	2	1	3	1	1	4	3	3	3	2	1	1	1	2	1	2	4	1	2	2	4	3	1	3	3
3	3	4	3	4	2	3	4	4	2	1	3	1	1	4	3	3	3	2	1	1	1	2	1	2	4	1	2	2	4	3	1	3	3
2	2	3	2	3	1	2	3	3	1	4	2	4	4	3	2	2	2	1	4	4	4	1	4	1	3	4	1	1	3	2	4	2	2
1	1	2	1	2	4	1	2	2	4	3	1	3	3	2	1	1	1	4	3	3	3	4	3	4	2	3	4	4	2	1	3	1	1
1	1	2	1	2	4	1	2	2	4	3	1	3	3	2	1	1	1	4	3	3	3	4	3	4	2	3	4	4	2	1	3	1	1
1	1	2	1	2	4	1	2	2	4	3	1	3	3	2	1	1	1	4	3	3	3	4	3	4	2	3	4	4	2	1	3	1	1
2	2	3	2	3	1	2	3	3	1	4	2	4	4	3	2	2	2	1	4	4	4	1	4	1	3	4	1	1	3	2	4	2	2
1	1	2	1	2	4	1	2	2	4	3	1	3	3	2	1	1	1	4	3	3	3	4	3	4	2	3	4	4	2	1	3	1	1
2	2	3	2	3	1	2	3	3	1	4	2	4	4	3	2	2	2	1	4	4	4	1	4	1	3	4	1	1	3	2	4	2	2
4	4	1	4	1	3	4	1	1	3	2	4	2	2	1	4	4	4	3	2	2	2	3	2	3	1	2	3	3	1	4	2	4	4
1	1	2	1	2	4	1	2	2	4	3	1	3	3	2	1	1	1	4	3	3	3	4	3	4	2	3	4	4	2	1	3	1	1
2	2	3	2	3	1	2	3	3	1	4	2	4	4	3	2	2	2	1	4	4	4	1	4	1	3	4	1	1	3	2	4	2	2
2	2	3	2	3	1	2	3	3	1	4	2	4	4	3	2	2	2	1	4	4	4	1	4	1	3	4	1	1	3	2	4	2	2

그림 8. 정방형으로 변형된 완벽한 암호행렬의 예

렬의 서로 같은 부분을 겹쳐서 결합하면, 즉, $\aleph^{v}_{(2)}(M^1)\oplus \aleph^{v}_{(7)}(M^2)$ 는 그림 8과 같은 (32+2)×(32+2) 의 형태를 갖는 정방형의 그리고 완벽한 암호행렬을 구성하게 된다.

5.2 정방형 암호행렬의 균일성 검토

1) 1차원적 균일성의 검토

1차원적 균일성의 검토는 런 테스트(runs up-and-down test)를 통해 수행된다. 길이가 N 인 수열에서 원소의 분포가 균일하다면 길이가 k 인 런이 발생할 이론적인 기대치는 $k < N-1$ 에 대하여

$$\frac{2[(k^2+3k+1)N-(k^3+3k^2-k-4)]}{(k+3)!} \quad (1)$$

이며 전체 런의 개수에 대한 기대치는 $(2N-1)/3$ 이다 [7]. 이러한 런 발생에 대한 이론적인 기대치와 수열로부터 계산된 런 값들을 비교하여 수열 내의 원소들이 균일하게 분포되었는지를 검토할 수 있다.

이를 일정한 위험수준을 갖는 통계적인 방법으로 처리하기 위해서는 다음의 정규분포를 따르는 Z 값을 계산하여 일정한 위험수준 α하에서 균일성을 검정할 수 있다. 이러한 통계치는 런의 길이와 무관하게 전체 런의 개수에 근거하여 런의 개수가 적으면 즉, 특정 런의 길이가 길어지면 수열을 이루고 있는 원소들이 균일한 분포로부터 추출되었다는 귀무가설을 받아들일 수 없게 됨으로써 검정을 하게 된다 [8].

$$Z=\frac{U-(2N-1)/3}{[(16N-29)/90]^{1/2}} \quad (2)$$

런 테스트를 통해 기본수열과 도약수열 내의 원소의 분포가 균일하다고 결론 내릴 수 있으면 그로부터 만들어지는 정방형의 행렬 또한 균일성을 갖는다는 가정을 한다.

a) 기본수열의 균일성 검토

정방형 행렬의 기본수열에서 암호구성에 사용되지 않는 양끝의 한 원소씩을 제외한 나머지 수열 (2 3 2 3 1 2 3 3 1 4 2 4 4 3 2 2 2 1 4 4 4 1 4 1 3 4 1 1 3 2 4 2)에 대하여 런을 작성하면 (1 0 1 0 1 1 1 0 1 0 1 1 0 0 1 1 0 1 1 1 0 1 0 1 1 0 1 1 0 1 0)이 되고 식(1)에 의해 다음 표 1과 같은 결과를 얻는다.

표 1. 기본수열에 대한 런 테스트 결과

구분 \ 런의 수	1	2	3	전체 런의 수
관찰치	15	5	2	22
이론치	13.42	5.63	1.56	21

표에서 보듯이 기본수열을 구성하는 각 원소는 균일하게 분포되어 있음을 아는데, 이러한 결과를 식 (2)를 사용하여 다른 각도에서 검토한다. 기본수열로부터 계산된 Z 값은 0.432 이고 $Z_{0.025}$=1.96 이므로 α=0.05 위험수준에서 기본수열은 균일성을 갖는다라고 결론을 내릴 수 있다.

b) 도약수열의 균일성 검토

정방형 행렬의 도약수열에서 맨 끝의 한 원소를 제외한 나머지의 수열 (4 1 3 1 2 1 1 4 2 3 2 2 4 3 3 4 4 2 3 2 2 4 3 3 4 4 1 3 1 2 1 1) 에 대하여 균일성을 검토한다. 이 수열에 대한 런을 작성하면 (0 1 0 1 0 1 1 0 1 0 1 1 0 1 1 1 0 1 0 1 1 0 1 1 1 0 1 0 1 0 1 1)이 되고 식(1)에 의해 다음 표 2와 같은 결과를 얻게 된다.

표 2. 도약수열에 대한 런 테스트 결과

구분 \ 런의 수	1	2	3	전체 런의 수
관찰치	18	4	2	24
이론치	13.8	5.82	1.61	21.67

표에서 보듯이 기본수열을 구성하는 각 원소는 균일하게 분포되어 있음을 아는데, 이러한 결과를 식(2)를 사용하여 Z 값을 계산하면 0.99 이고 $Z_{0.025}$=1.96 이므로 α=0.05 위험수준에서 도약수열은 균일성을 갖는다라고 결론을 내린다.

2) 2차원적 균일성의 검토

정방형 행렬 내의 각 원소들에 대한 2차원 평면상의 균일성을 검정하기 위하여 그림 7의 암호행렬에서 가장 바깥쪽 원소들을 제외한 중심원소들로 이루어진 32×32 행렬을 고려한다. 이렇게 하는 이유는 이 원소들만이 암호

행렬에서 암호로 사용되기 때문이며 이 행렬을 중심행렬이라 부르기로 한다. 이러한 중심행렬을 일정한 크기를 갖는 4개의 부분행렬로 나누고, 행렬 내 원소의 위치를 2차원 평면상의 좌표 값에 대응시키기 위하여 일반적으로 사용되는 2차원 배열 형태의 인덱싱(indexing) 방법을 그대로 좌표 값으로 사용한다. 즉, 원래의 정방형 암호행렬의 첫 행과 첫 열의 원소를 (0, 0)으로, 첫 행과 둘째 열의 원소를 (1, 0), 첫 열과 둘째 행의 원소를 (0, 1) 등으로 하면 중심행렬의 4개의 부분행렬은 중심원소의 전체 범위인 (1, 1) - (32, 32)를 4개의 범위로 나눈 (1, 1) - (16, 16), (1, 17) - (17, 32), (17, 1) - (32, 16), (17, 17) - (32, 32)가 된다.

이러한 각 4개의 구간에는 256개의 원소가 있게 되는데, 귀무가설을 각 원소가 균일분포로부터 추출되었다 라고 두면 각 구간에서 이론적으로 예상되는 각 원소의 발생 수는 64가 된다. 이러한 이론적인 기대치와 실제로 관측된 각 구간의 각 원소의 발생수로부터 카이제곱 값을 산출할 수 있는데 표 3에 그 결과가 정리되어 있다.

표 3. 카이제곱 적합도 검정을 이용한 정방형 행렬의 2차원적 균일성 검토

원소 / 구간	1	2	3	4	이론적 기대치
(1,1) -(16,16)	54	65	72	65	64
(1,17) -(17,32)	72	65	54	65	64
(17,1) -(32,16)	65	72	65	54	64
(17,17)-(32,32)	65	54	65	72	64
χ_0^2	2.59	2.59	2.59	2.59	

표 3으로부터 각 원소의 χ_0^2 값은 공히 2.59가 됨을 알 수 있고 α=0.05 의 위험수준에서 $\chi^2_{.05,3}$=7.81 이므로 $\chi_0^2 < \chi^2_{.05,3}$ 이 성립되어 귀무가설을 버릴 수 없게 된다. 따라서 각 원소는 2차원 행렬 내에서 균일하게 분포되어 있다고 결론을 내린다.

5.3 완벽하고 균일한 정방형 행렬의 응용 예

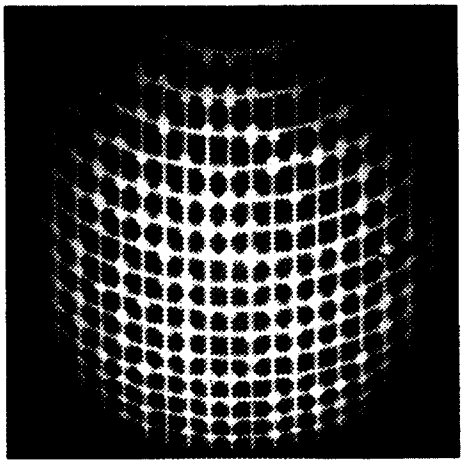
형태가 정방형이고 구성원소가 균일하게 배열된 암호행렬의 응용 범위는 광범위하리라 생각된다. 전술한 바와 같이 머신비젼분야, 특히 암호행렬을 구조화된 광원의 형태로 이용하는 삼차원 좌표 측정 분야에서는 직사각형의 형태를 갖는 암호행렬보다 측정 대상 물체의 형태 제한을 덜 받을 수 있다는 장점을 갖는다. 본 논문에서는 정방형 행렬이 갖는 원소의 균일성을 이용하여 영상처리를 수월하게 할 수 있는 예를 보이고자 한다.

영상처리(image processing)는 영상으로부터 필요로 하는 정보를 얻기 위하여 주어진 영상을 인위적으로 처리하는 과정인데, 이러한 처리과정은 크게 전 처리 과정과 후처리 과정으로 나뉘어진다. 일반적으로 머신비젼등에서 카메라와 같은 영상 획득 장비로부터 얻어지는 영상은 필요로 하는 정보 이외의 다른 영상정보나 잡음(noise)등을 포함하고 있기 때문에 전처리를 통하여 필요한 정보를 추출하기 위한 준비작업을 하게 된다.

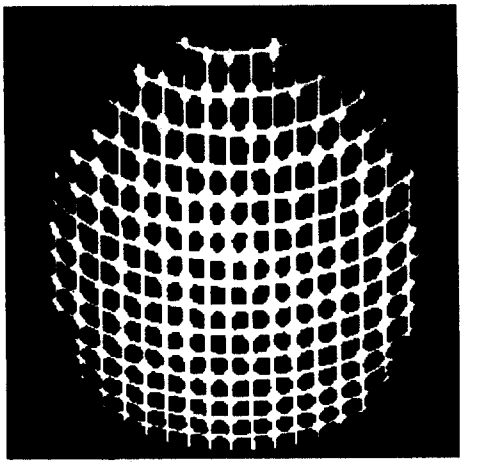
이러한 전처리는 필요에 따라 여러 가지의 과정을 거칠 수 있는데, 한 예로 이치화(binarization)를 들 수 있다. 이치화는 다수의 명암을 갖고 있는 영상을 흑과 백의 두 가지 명암을 갖는 영상으로 만드는 과정으로 이치 화를 통해 영상에서 필요로 하는 부분을 제외한 나머지 영상을 효과적으로 줄일 수만 있다면 이치화된 영상이 다수의 명암을 갖고 있는 영상보다 처리에 있어 훨씬 수월하다. 따라서 효과적인 이치화의 관건은 가능한 영상에서 필요 부분을 많이 보존하고 배경등 필요 없는 부분은 많이 줄이는 것이다.

그림 9의 (a)는 카메라로부터 얻어진 영상으로 공(ball)에 비추어진 무늬를 보여주고 있으며 256개의 명암 치를 갖고 있다. 이 영상에서 무늬의 정보가 중요하다고 할 때 무늬의 정보를 가능한 보존하면서 이치화하는 것이 필요하다. 영상을 이치화하는 방법 중 하나는 0과 255 사이의 적절한 명암치 하나를 선택하여 이 역치(threshold)보다 큰 명암치를 갖는 부분은 백으로 나머지는 흑으로 처리하는 것이다. 하나의 역치를 이용하는 이 방법을 사용하여 최초의 영상을 이치화한 결과가 그림 9의 (b)에 나타나 있다. 이 경우 역치는 인위적으로 선택되어 영상에서 필요한 부분(무늬 부분)을 최대한 유지하도록 하였으나 공 주변의 무늬를 많이 잃고 있음을 관찰할 수 있다.

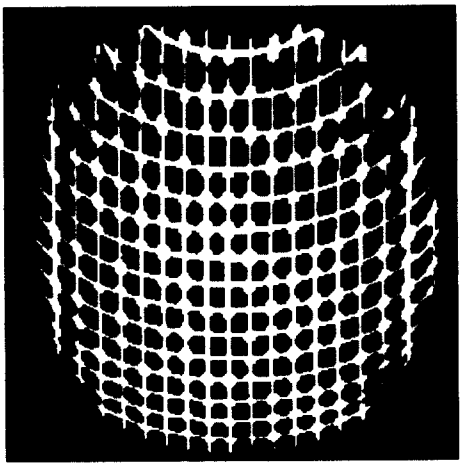
하나의 역치를 사용하는 것보다 영상의 부분적인 명암치 변화에 따라 적절한 역치를 찾는 적응적(adaptive) 방법이 효과적일 수가 있는데 이러한 방법 중 하나가 [3]에



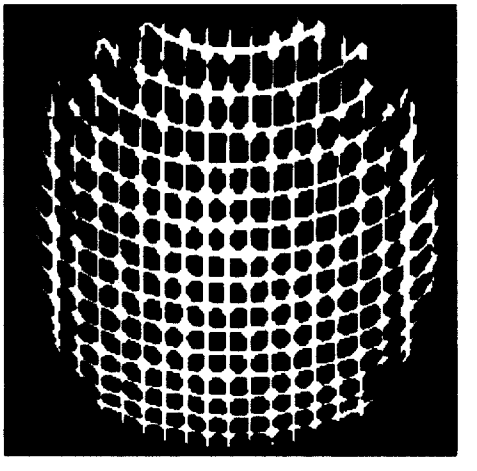
(a) 카메라에서 얻어진 초기 영상



(b) 하나의 역치를 이용한 이치화



(c) 적응적 방법에 의한 이치화



(d) 균일한 행렬을 이용한 이치화

그림 9. 완벽하고 균일한 암호행렬이 이치화(binarization)에 적용된 예

소개된 바 있다. [3]에서는 대상이 되는 영상을 그보다 작은 정사각형의 창(window)을 통해 본 작은 영상부위로 나누어 각 부위에 가장 적절한 역치값을 자동으로 부여한다. 이 방법을 이용하여 이치화된 영상의 예가 그림 9의 (c)에 나타나 있는데, (b)와 비교해 볼 때 원래 영상의 무늬를 더욱 잘 표현하고 있음을 알 수 있다. 그러나 이 경우 각 부위의 역치를 계산하기 위하여 정사각형의 창에 내포되어 있는 모든 화소(pixel)의 명암치를 사용하므로 만약 모든 화소를 사용하지 않고도 동일한 이치화의 효과를 얻을 수만 있다면 더욱 효율적일 것임이 분명하다.

동일한 영상에 대해 본 논문에서 개발한 정방형의 균일한 행렬을 이용하여 선택적인 화소의 명암치만을 역치의 계산에 사용하여 처리한 결과가 그림 9의 (d)에 나타나 있다. 즉, 균일한 암호행렬 내의 각 원소의 위치는 2차원 평면상에서 균일하게 분포되어 있으므로 특정 원소

(예에서는 3)의 위치에 해당하는 화소만을 선택하여 계산에 이용하였다. 그 결과 전체의 화소를 이용한 결과와 동일한 이치화의 효과를 얻을 수 있었고, 다른 원소를 선택해도 유사한 결과를 얻을 수 있음을 확인할 수 있었다.

6. 결 론

본 논문에서는 삼차원 비젼 시스템의 개발 시 상응문제를 해결하기 위해 고안되었던 암호행렬에 대해 그 이론적인 특성을 살펴보고자 하였다. 이를 위해 우선 암호행렬의 생성에 사용된 서로 맞물린 수열의 성질과 이러한 수열들로부터 암호행렬을 만드는 연산에 대해 간략히 알아보았고, 생성된 암호행렬이 갖는 이론적인 특성을 규명하였다. 또한 이러한 암호행렬로부터 읽혀지는 암호의 특성을 살펴봄으로써 암호행렬의 응용 시 사용될 수 있는 이론적인 배경을 제공하였다. 규명된 이론적인 특성은 암호행렬의 형태변화를 줄 수 있는 변환함수의 개발을 가능하게 하는데 본 논문에서는 이러한 변환함수를 개발하고 이를 사용하여 암호행렬 내의 원소가 균일한 형태로 배열되는 정방형의 행렬을 구성해 보았다. 암호행렬의 응용 예로서 이러한 균일한 정방형의 행렬을 영상처리의 이치화(binarization) 과정에 적용시켜 보았으며 그 결과 이차원적인 컨벌루션(convolution)등 평면의 좌표 값에 의존하는 계산이 필요한 경우 균일한 암호행렬을 이용하여 샘플링을 통한 효율적인 계산을 할 수 있음을 볼 수 있었다.

암호행렬의 이론적인 특성을 배경으로 한 암호행렬의 적절한 응용은 예에서와 같이 영상 및 암호처리 등의 분야는 물론 관련되는 여러 분야에서 발견될 수 있으리라 기대된다. 따라서 암호행렬의 응용분야를 확장하는 일은 앞으로의 과제로 남는다.

참고문헌

[1] Griffin, P. M., Narasimhan, L. S., and Yee, S. R., "Generation of uniquely encoded light patterns for range data acquisition," *Pattern Recognition*, vol.25, no.6, pp.609-616, 1992.

[2] Griffin, P. M. and Yee, S. R., "The use of a uniquely encoded light pattern for range data acquisition," *Proc. 13th Annual Conf. Computers & Industrial Engineering*, vol.21, nos.1-4, pp.359-363, 1991.

[3] Yee, S. R. and Griffin, P. M., "Three-dimensional imaging system," *Optical Engineering*, vol.33, no.6, pp.2070-2075, 1994.

[4] Yee, S. R., "Interlocking Number String and Code Matrix," *Computers & industrial Engineering*, vol.31, no.3/4, pp.929-932, 1996.

[5] Yee, S. R., "Interlocking Number String and Code Matrix: Transformation," *Proc. of ICC & IE 96*, vol. 1, pp.253-256, 1996.

[6] 이성룡, "유일하게 구성된 암호행렬의 균일성에 관한 연구," 한국외국어대학교 논문집, vol.29, pp.259-269, 1996.

[7] Olmstead, P. S., "Distribution of sample arrangements for runs up and down," *Annals of Mathematical Statistics*, vol.17, pp.24-33, 1946.

[8] Levene, M., "On the power function of tests of randomness based on runs up and down," *Annals of Mathematical Statistics*, vol.23, pp.34-56, 1952.

97년 9월 최초 접수, 97년 11월 최종 수정