

무손실 이진 영상 압축을 위한 컨텍스트 기반 계층적 열거 부호화

Context-Based Hierarchical Enumerative Coding for Lossless Bi-level Image Compression

임재혁, 정제창

한양대학교 전자통신전파공학과

본 논문에서는 컨텍스트 기반 계층적 열거 부호화를 이용한 무손실 이진 영상 압축 알고리즘을 제안한다. 이진 영상내에 존재하는 인접한 화소간의 상호상관성을 이용하여 이진 영상을 1차원의 수열로 재구성하고, 이에 대해 계층적 열거 부호화를 실행한다. 제안하는 알고리즘은 덧셈 및 비교 연산만으로 구현이 가능하므로 그 복잡도가 매우 낮을 뿐만 아니라, CCITT 테스트 영상을 대상으로한 부호화 성능 실험에서 우수한 성능을 나타낸다. 부호화 성능 비교에서 이진 영상 부호화 국제표준인 JBIG, G3, G4 및 GIF에 비해 우수한 압축 성능을 보인다.

I. 서론

대부분의 무손실 압축기법들은 입력 데이터가 데이터 전체 혹은 넓은 범위에서 각각 심볼의 발생 확률의 통계적 특성이 변하지 않는다는 가정하에 그 확률의 편중을 이용하여 압축을 한다. 따라서 입력 데이터의 통계적 특성이 시간에 따라 변하게 되면 그 성능이 급격히 떨어지게 된다. 이러한 시간에 따른 통계적 특성의 변화에 효과적으로 적용하기 위해 적응적 방법들이 많이 제안되었다. [1][2]

Lynch[3] 와 Davisson[4]에 의해 고안된 열거 부호화(enumerative coding) 기법 또한 데이터의 통계적 특성의 시간적 변화에 적응적으로 부호화 할 수 있도록한 적응적 압축 기법이다. 열거 부호화 기법은 그 방법이 매우 간단할뿐만 아니라, 입력 이진 수열을 작은 블록으로 분할하여 처리하므로 블록 크기를 작게 할 경우 통계적 특성의 부분적 변화에도 적응적으로 부호화 할 수 있다.

본 논문에서는 컨텍스트 기반 계층적 열거 부호화를 이용한 무손실 이진 영상 압축 알고리즘을 제안한다. 이진 영상내에 존재하는 인접한 화소간의 상호상관성을 이용하여 이진 영상을 1차원의 수열로 재구성하고, 이를 계층적 열거 부호화를 실행한다. 본 논문의 구성은

다음과 같다. II장에서 이진 벡터의 열거 부호화 방법에 대해서 살펴보고, III장에서는 이를 좀더 확장하여 음이진 정수 벡터의 열거 부호화 방법에 대해서 살펴본다. IV장에서는 계층적 열거 부호화 방법에 대해서 살펴보고, V장에서는 계층적 열거부호화 방법을 보다 효과적으로 무손실 이진 영상 압축에 적용하기 위해서 컨텍스트 기반의 알고리즘에 대해서 설명한다. VI장에서 제안된 부호화기의 CCITT 테스트 영상에 대한 부호화 결과 및 기존의 이진 영상 부호화 국제표준인 JBIG[5], G3[6], G4[7] 및 GIF와 그 성능을 비교하며, VII장에서 결론을 맺는다.

II. 이진벡터(binary vector)의 열거부호화

n 개의 0 혹은 1을 원소로 갖고, 원소들의 합이 w 인 이진벡터를 생각해보자. 만약 w 값을 알고 있다고 가정하면, 앞서의 조건을 만족하는 벡터는 모두 $C(n, w) = \frac{n!}{w!(n-w)!}$ 만큼 존재한다. 이때, 이 벡터들을 오름차순으로 정렬했을 때 부호화 하고자하는 벡터가 몇번째에 위치하는지를 부호화함으로써 이진 벡터를 유일하게 나타낼수 있다. 따라서 열거 부호화 기는, n 값이 미리 정해진 값이라고 할 때, w

값 및 벡터의 순번을 보냄으로써 이진 벡터를 부호화한다. 이때 벡터의 순번을 부호화하는데에는 $\lceil \log_2 C(n, w) \rceil$ 비트가 요구된다.

표 1. n=6, w=2 를 만족하는 이진 벡터들의 오름차순 정렬

순번	벡터
0	000011
1	000101
2	000110
3	001001
4	001010
5	001100
6	010001
7	010010
8	010100
9	011000
10	100001
11	100010
12	100100
13	101000
14	110000

n=6 이고 w=2 인 벡터 001010을 부호화하는 경우를 생각해 보자. 표 1은 전체 원소의 개수가 6이고, 그중 2개의 1을 가진 이진 벡터의 가능한 모든 조합을 오름차순으로 정리한 것이다. 우리가 부호화 하고자 하는 벡터 001010의 순번은 4이므로, 이것을

$\lceil \log_2 C(6,2) \rceil = 4$ 비트로 표현, 즉 0100을 나타내면 부호화가 이루어지게 된다.

부호화 및 복호화 과정에서 사용되는 대부분의 연산은 부호화하고자 하는 벡터의 순번을 찾아내는데 사용되는데, 여기에는 미리 계산된 바이노미얼 계수(binomial coefficients)를 look-up table 형태로 참조하여 빠르게 찾아낼 수 있는 알고리즘이 존재하는데, 특별히 작은 n값에 대해 매우 빠르게 찾아낼 수 있다. 부호화기에서 순번을 찾아내고, 복호화기에서 순번을 이용해 벡터를 다시 생성해내는 과정은 다음과 같다.

부호화기

1. ordinal = 0, left ones = w 로 초기화
2. For i=0 to n-1
 - i 번째 숫자가 1 이면

- left ones = left ones - 1

- ordinal = ordinal +

$C(n-i-1, \text{left ones}+1)$

3. ordinal을 $\lceil \log_2 C(n, w) \rceil$ 비트를 사용하여 나타낸다.

위의 부호화 알고리즘을 사용하여 앞서 예로 든 벡터 001010을 부호화 하는 과정은 다음의 표2 와 같다.

표 2. 부호화 예

i	left ones	add	ordinal
2	1	$C(6-2-1, 1+1)$	3
4	0	$C(6-4-1, 0+1)$	4

복호화기

1. index = ordinal, left ones = w 로 초기화

2. For i=0 to n-1

- If index < $C(n-i-1, \text{left ones})$,
 - 출력벡터의 i 번째 숫자 = 0
- Else,
 - 출력벡터의 i 번째 숫자 = 1
 - index = index -

$C(n-i-1, \text{left ones})$

- left ones = left ones - 1

위의 복호화 알고리즘을 사용하여 앞서 부호화 예에서 부호화된 4를 복호화 하는 과정을 아래의 표3에서 나타내고 있다.

표 3. 복호화 예

i	index	compare	left ones	decoded bit
0	4	$C(6-0-1, 2)=10 > \text{index}$	2	0
1	4	$C(6-1-1, 2)=6 > \text{index}$	2	0
2	4 4-3=1	$C(6-2-1, 2)=3 \leq \text{index}$	2 2-1=1	1
3	1	$C(6-3-1, 1)=2 > \text{index}$	1	0
4	1 1-1=0	$C(6-4-1, 1)=1 \leq \text{index}$	1 1-1=0	1
5	0	$C(6-5-1, 0)=1 > \text{index}$	0	0

부호화 및 복호화 예에서 볼 수 있듯이 열거 부호화기는 바이노미얼 계수가 미리 look-up table 형태로 제공되지만 하면 덧셈 및 비교연산만 가지고 모든 과정을 수행할 수 있으므로 복잡도가 매우 낮은 알고리즘이다.

이 방법은 이진 수열(binary sequence)의 부호화에 직접 응용할 수 있다. 즉 이진 수열을 크기 n 을 갖는 블록으로 나누고 각각을 이진 벡터로 보고 위의 방법을 적용하면 된다. 특히 작은 크기의 블록으로 나누어 처리하게 되면 신호의 통계적 특성의 부분적 변화를 매우 효과적으로 활용할 수 있다. 그러나 이 경우 각각의 블록에 대해 w 값을 부호화하는데 사용되는 부가정보가 점점 커지게 되는데, 이 값을 $\lceil \log_2(n+1) \rceil$ 의 비트로 고정장부호화(Fixed Length Coding)로 부호화 했을 경우 상당히 큰 크기를 차지하게 된다.

III. 음아닌 정수 벡터의 열거 부호화

p 개의 음아닌 정수를 원소로 갖고, 그 합이 q 인 벡터를 생각해보자. 만약 q 값을 알고 있다고 가정하면, 앞서의 조건을 만족하는 벡터는 모두

$H(p, q) = C(p+q-1, q) = C(p+q-1, p-1)$ 만큼 존재하게 되고, 이 벡터들을 오름차순으로 정렬했을 때 임의의 벡터의 순번으로 그 벡터를 유일하게 나타낼 수 있다. 이때 벡터의 순번은 $\lceil \log_2 C(p+q-1, q-1) \rceil$ 의 비트를 사용하여 부호화 할 수 있다.

이러한 방법은 앞서의 이진 벡터의 경우와 연관지어 생각할 경우 벡터의 원소값을 연속된 0의 나열로 나타내고, 각각의 원소를 $p-1$ 개의 1로써 구분하는 경우로 볼 수 있다. 즉 $p+q-1$ 개의 원소를 갖고, 그 중 1의 개수가 $p-1$ 개인 이진 벡터의 열거 부호화라 생각할 수 있다.

IV. 계층적 열거 부호화

II장에서 설명한 것처럼 주어진 이진 벡터를 열거 부호화하기 위해서는 원소의 합 w 및 그 벡터의 순번을 나타내면 된다. 그러나 w 값을 부호화하는데 FLC를 사용할 경우 여기에

사용되는 부가정보가 너무 커지므로 효율적인 부호화가 이루어지기 어렵다. 이러한 문제는 신호의 통계적 특성의 부분적 변화를 효과적으로 활용하기 위해 블록의 크기를 작게 하여 부호화할 경우 그 심각성이 더욱 커지게 된다. Levent[8]는 w 값을 효과적으로 부호화하기 위해 계층적 방법을 제안하였는데, 여기에서 그 내용을 소개한다.

이진 수열을 부호화하고자 할 때, 이를 작은 블록으로 나누고, 각각을 이진 열거 부호화를 이용해 부호화하는 경우를 생각해보자. 일단 부호화기와 복호화기 모두에서 블록안의 1의 개수(w)를 알게되면, 이 블록(벡터)의 순번을 부호화하므로써 각각의 블록을 나타낼 수 있다. 따라서 블록의 순번을 부호화하기전에 먼저 1의 개수를 보내야만 한다. 여기서 다시 연속된 블록의 그룹을 생각해보자. 만약 그룹안의 1의 개수를 안다고 가정하면, 각각의 블록내의 합(1의 개수)은 앞서 III장에서 설명한 음아닌 정수 벡터의 열거 부호화를 이용하여 부호화 할 수 있다. 그룹내의 1의 개수는 연속된 그룹의 그룹(group of group)의 합을 안다고 가정하고, 앞서의 방법과 같은 방법으로 부호화 할 수 있다. 이와 같은 과정을 반복적으로 적용함으로써, 전체의 이진 수열을 계층적 열거 부호화를 행할 수 있다.

입력 이진 수열 x 를 생각해보자. x 의 j 번째 숫자를 $x(j)$ 라 하자. 앞서의 계층적 부호화 방법은 입력 이진 수열 x 에 대해 각각의 단계에서 작은 블록의 합을 계속적으로 만들어 나가는 sum-up tree 형태로 이해될 수 있다. 이 sum-up tree를 구성하는 방법은 다음의 식으로 나타내어질 수 있다.

$$w_{0,i} = \sum_{j=i \cdot n_0}^{(i+1) \cdot n_0 - 1} x(j) \quad (1)$$

$$w_{p,i} = \sum_{j=i \cdot n_p}^{(i+1) \cdot n_p - 1} w_{p-1,j}, \quad p > 0 \quad (2)$$

위 식에서 n_p 는 p 번째 단계에서의 블록 혹은 그룹의 크기를 나타낸다. 위의 트리를 구성해 나가는 과정을 그림 1에서 나타내고 있다. 계층적 열거 부호화는 트리의 맨 상위 단계부터 순차적으로 부호화 하게 된다. 맨 상위의

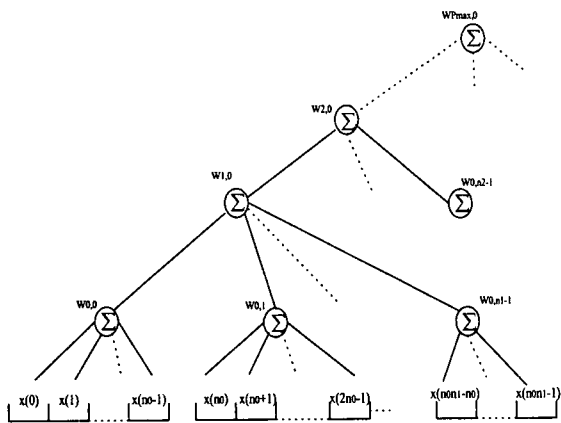


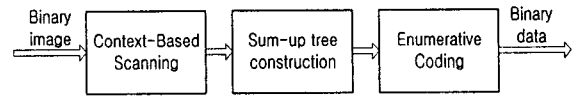
그림 1. Sum-up tree

$w_{p_{max},0}$ 는 FLC를 이용하여 부호화하고, 맨 하위 단계는 이진 벡터의 열거 부호화를 사용하여 부호화하며, 그 사이 단계는 음이진 정수 벡터의 열거부호화를 이용하여 부호화하게 된다.

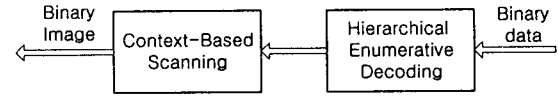
V. 컨텍스트 기반 열거 부호화를 이용한 무손실 이진 영상 압축

대부분의 무손실 압축기법들은 입력 데이터가 데이터 전체 혹은 넓은 범위에서 각각 심볼의 발생 확률의 통계적 특성이 변하지 않는다는 가정하에 그 확률의 편중을 이용하여 압축을 한다. 따라서 입력 데이터의 통계적 특성이 시간에 따라 변하게 되면 그 성능이 급격히 떨어지게 된다. 이러한 시간에 따른 통계적 특성의 변화에 효과적으로 적응하기 위해 적응적 방법들이 많이 제안되었는데, 앞서 설명된 계층적 열거 부호화에서는 입력 데이터를 작은 블록으로 나누어 부호화를 하기 때문에 신호의 시간에 따른 통계적 특성의 변화에도 적응적으로 부호화를 행할 수 있다.

그림 2는 제안하는 이진 영상 부호화기의 블록 다이어그램이다. 이진 영상을 계층적 열거 부호화하기 위해서는 먼저 입력 영상을 1차원의 이진 수열로 재구성해야 한다. 여기에 가장 간단한 방법으로는 영상의 좌측 상단부터 좌측에서 우측으로, 상단에서 하단으로 순차적으로 스캔하는 방법을 생각할 수 있다. 하지만 이와 같은 방법으로 얻어진 이진 수열은 각각의 위치에 따라서 0과 1의 통계적 발생



(a)



(b)

그림 2. 제안하는 이진 영상 부호화기 및 복호화기 블록 다이어그램 (a) 부호화기 블록 다이어그램 (b) 복호화기 블록 다이어그램

빈도가 급격히 변하므로 엔트로피 부호화했을 경우 그리 좋은 결과를 얻을 수 없다. 따라서 효과적인 부호화를 위해서는 얻어진 1차원의 이진 수열의 통계적 특성이 각각의 구간에서 최대한 0 혹은 1, 한쪽으로 편중되고 그 특성이 최대한 변화가 없어야 한다.

이와 같은 1차원 이진 수열을 얻기 위해서 본 논문에서는 영상의 화소값을 이용하여 컨텍스트를 구성하여 이진 영상을 1차원의 이진 수열로 만든다. 즉 현재 스캔하는 화소의 좌측 혹은 상위 라인의 k개의 화소값을 참고하여 2^k 의 집합중의 하나에 해당 화소를 등록시킨다.

			X7	X6	
	X5	X4	X3	X2	
	X1	X0	C		

그림 3. k=8 일때의 컨텍스트 구성을 위한 주위 화소

그림 3에서 처럼 현재 화소 C를 스캔하는 경우

$$SetIndex = \sum_{i=0}^{k-1} x_i \cdot 2^i \quad (3)$$

를 계산하여 해당하는 집합에 현재 화소 C를 등록시킨다. 이와 같은 방법으로 영상 전체에 대해서 각각의 화소를 2^k 개의 집합으로 분류할 수 있게 된다. 여기서는 casual 컨텍스트를 사용하였으므로 별도의 부가정보를 복호화기에 전해주지 않아도 복호기에서 다시 역과

정을 행할수 있게 된다. 이렇게 얻어진 각각의 1차원 이진 수열 집합을 하나의 이진 수열로 묶고 이것을 앞서 설명한 계층적 열거 부호화기의 입력 이진 수열로 넣어주어 부호화를 행한다.

VI. 실험 결과

제안한 부호화기의 압축 성능평가를 위해 CCITT 테스트 영상에 대해서 부호화 실험을 수행하였다. 각각의 테스트 영상들은 1728×2376 크기를 갖는 이진 영상이다. 사용한 부호화기에서는 계층적 열거 부호화를 위해 각각의 단계에서의 블록의 크기를 $n_0=64$, $n_1=8$, $n_2=4$, and $n_p=1$ for $p > 2$ 로 사용하였다. 또한 컨텍스트를 구성하기 위한 파라미터 k 는 8을 사용하였다. 주위 화소 8개의 구성은 그림 3과 같다. 압축된 데이터에는 영상의 가로, 세로 크기 정보 및 기타 정보를 위한 5 bytes의 부가정보와 2^k 개의 집합 각각의 크기 정보를 함께 포함하고 있다.

또한 제안한 부호화기의 성능을 다른 부호화 방식들과 비교하기 위하여 이진 영상 부호화 국제표준인 JBIG, G3, G4 및 GIF로 동일한 영상에 대해 부호화를 실행하여 그 결과를 비교하였다. 표 4는 CCITT 테스트 영상들에 대한 각각의 부호화기 및 제안한 부호화기의 성능을 나타내고 있다. 그림 4는 실험에 사용한 CCITT 테스트 영상을 나열하였다.

실험 결과에서 볼수 있듯이 제안된 부호화기는 GIF, G3, G4에 비교했을 경우 모든 테스트 영상에 대해 우수한 결과를 보이고 있으며, JBIG과 비교할 때 테스트 영상에 따라서 약간의 차이는 있으나 대부분의 영상에서 근소한 차이로 우수한 성능을 보이고 있다.

VII. 결론

본 논문에서는 컨텍스트 기반 계층적 열거 부호화를 이용한 무손실 이진 영상 압축 알고리즘을 제안하였다. 이진 영상을 주위의 k 개의 화소를 이용하여 컨텍스트를 구성하고, 이를 이용하여 2^k 개의 1차원 이진 수열을 만

표 4. CCITT 테스트 영상에 대한 부호화 결과. 원 영상 데이터 크기 = 513216 bytes (unit : bytes)

Image	GIF	G3	G4	JBIG	CHEC
CCITT 1	42,564	38,815	18,680	16,830	15,458
CCITT 2	36,493	35,835	11,321	8,958	9,581
CCITT 3	71,625	66,491	29,239	23,642	22,822
CCITT 4	126,188	109,523	69,927	58,748	56,195
CCITT 5	76,914	69,796	32,734	28,092	26,854
CCITT 6	54,727	52,625	17,227	13,503	13,724
CCITT 7	127,634	107,868	69,914	60,640	58,629
CCITT 8	64,290	64,342	19,703	15,135	15,314

* CHEC (Context-based Hierarchical Enumerative Coding) : 제안된 부호화기

들었다. 이렇게 만들어진 이진 수열을 계층적으로 열거 부호화를 행하여 신호내에 존재하는 부호의 발생확률의 편중을 효과적으로 활용하고, 통계적 특성의 시간적 변화에도 효과적으로 적응하여 부호화하도록 하였다. 실험 결과에서 나타나듯이 제안된 부호화기는 그 성능이 매우 우수하고, 기존의 이진 영상 압축 방법들과 비교했을 때 더 나은 압축 성능을 보이고 있다. 또한 열거 부호화 방법은, 바이노미얼 계수를 look-up table 형태로 참조하여 덧셈 및 비교 연산만을 가지고 모든 부호화 과정을 처리하므로 그 복잡도가 매우 낮은 장점을 가지고 있다.

VIII. 참고 문헌

- [1] J. Rissanen, "A universal data compression system." *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 656-664, September 1983.
- [2] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon and R. B. Arps, "An overview of the basic principles of the Q-coder adaptive binary arithmetic coder," *IBM*

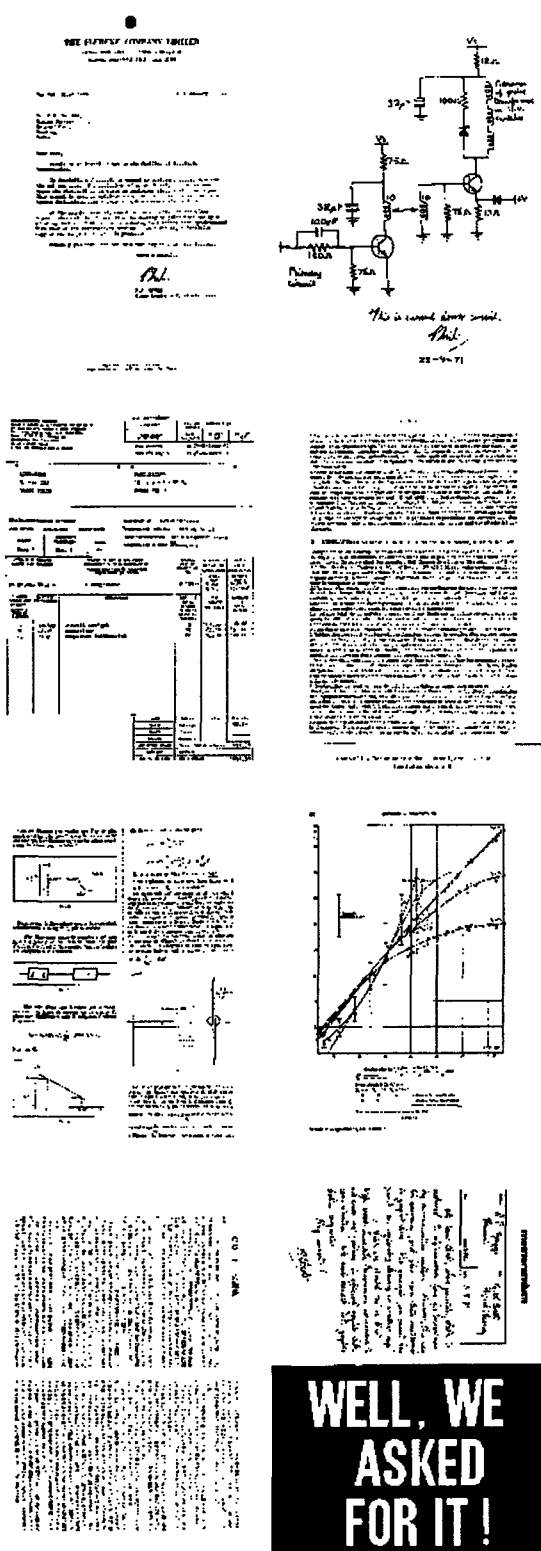


그림 4. 실험에 사용한 CCITT 테스트 영상.
좌측상단부터 좌에서 우로 상단에서 하단으로
차례로 표4 에서 나타냄

Journal of Research and Development, vol. 32, no. 6, pp.717-726, November 1988.

[3] T. J. Lynch, "Sequence time coding for data compression," *Proceedings of the IEEE*, vol. 54, no. 10, pp. 1490-1491, October 1966.

[4] L. D. Davisson, "Comments on 'Sequence time coding for data compression'," *Proceedings of the IEEE*, vol. 54, no. 12, pp. 2010, December 1966.

[5] ITU-T Rec. T.82 - Information Technology - Coded Representation of Picture and Audio Information - Progressive Bi-level Image Compression, 1993.

[6] ITU-T Rec. T.4 Standardization of Group 3 Facsimile Apparatus for Document Transmission, 1984.

[7] ITU-T Rec. T.6 Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus, 1988.

[8] L. Oktem, J. Astola, "Hierarchical Enumerative Coding of Locally Stationary Binary Data," *Electronics Letters*, vol. 35, no. 17, pp. 1428-1429, August 1999.