論文 98-23-7-19

고속 전송을 위한 V.42bis 데이터 압축 기법의 개선

정회원 조 성 렬*, 최 혁**, 김 태 영**, 김 태 정**

Data Compresson for High Speed Data Transmission

SungRyul Cho*, Hyuk Choi**, TaeYoung Kim**, Tae Jeong Kim** *Regular Members*

*이 논문은 한국학술진흥재단으로부터 지원을 받은 연구의 결과중 일부임.

요 약

이 논문에서는 비동기식 데이터 압축의 국제 표준으로 되어있는 Lempel-Ziv-Welch 부호의 일종인 V.42bis 방식을 데이터의 고속 전송에 적용할 경우 압축 과정에서 나타나는 여러 현상들을 분석하고 이에 맞는 변형기법을 제안한다. 제안된 기법은 압축률을 결정하는 중요한 요인중의 하나인 부호책의 크기를 최적화하고, 부호책의 갱신 방법을 개선하여 압축률을 향상시킨다. 또 빈번한 압축 형식 전환에서 오는 문제점을 분석하고 형식 전환에 사용되는 문턱값 조절로 이를 어느정도 개선하여, 압축률의 시간에 따른 변화를 줄인다는 측면에서 성능 향상을 이루었다.후자의 개선은 데이터의 고속 전송시에 완충기(buffer) 설계 및 제어에 중요한 기여를 한다.

ABSTRACT

V.42bis, a type of LZW(Lempel-Ziv-Welch) code, is well-known as theinter national standard in asynchronous data compression.In this paper, we analyze several undesirable phenomena arising from the application of v.42bis to high speed data transmission, and we propose a modified technique to overcome them.The proposed technique determines the proper size of the dictionary, one of important factors affecting the compression ratio, and improves the method of dictionary generation for a higher compression ratio. Furthermore, we analyze the problem of excessive mode changes and solve it to a certain degree by adjusting the threshold for mode change.By doing this, we can achieve smaller variation of the compression ratio in time. This improvement contributes to easier and better design and control of the buffer in high speed data transmission.

I. 서 론

주어진 전송로를 최대한 활용하면서 고속으로 데이터 전송을 수행하기위해서는 데이터의 압축이 필수적이다. 전통적으로 데이터 압축은 모뎀에서와 같이 비동기식 데이터 전송의 경우에 사용되어 왔으며, 1980년대 중반부터 고속 모뎀의 필요성이 증가되면서 일반 가입자 전화선에서 데이터의 비동기식 전송을 위한 국제 표준 권고안인 CCITT V.42bis가 결정되어 사용되고 있다[1]. 이 표준 권고안은 가변길이 데이터를 일정길이 부호로 압축하는Lempel-Ziv-Welch(LZW) 부호의 일종이다. 그러나 이 압축 방법에는 압축률이 고르지 못하고,빈번한 형식(mode) 전환에서 중복성(redundancy)이 부가될 뿐 아니라, 완충기(buffer) 넘침(overflow)이나 고갈(underflow)등의 여러가지 문제가 있어서 고속 데이터의 전송에 적합한

* 대우전자 디지털 TV 연구소
** 서울대학교 전기공학부
論文番號 : 97317-0906
接受日字 : 1997年 9月 6日

새로운 방법이 요구된다.일부 모뎀에서는 비동기식 전송 뿐만 아니라 동기식 전송방식을 사용하여 데이터의 처리량을 늘리고 있다[2].비동기식 전송은 바이트(byte) 단위로 전송되는 것을 말하며 동기식 전송은 HDLC/SDLC와 같이 프레임(frame) 단위로의 전송을 말한다[3]. 이들의 형태는 그림 1과 같다.

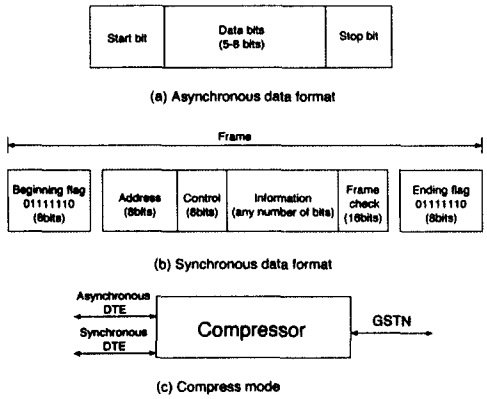


그림1. 데이터의 형태 및 압축 형식

동기식 데이터의 압축은 일부 복잡한 망(network)에서 사용되는라우터(router)나 다중장치(multiplexer)에서만 사용되어 왔으나 더욱 빠른데이터의 전송이 요구되고 비동기식 뿐만 아니라 동기식 데이터 전송을 함께할 수 있는 모뎀이 등장하는 환경에서 그 중요성이 크게 증대되었다.

동기식으로 데이터를 압축하여 전송할 때 압축기는 프레임의 경계를 인식할 수 있어야 하고 프레임의 내용만을 압축하여 전송하게 된다. 그러므로 프레임 안에서는 부호들 사이에 쉬는 시간이 없어야 하며 프레임간 처리 지연(throughput delay)없이 압축 전송되어야 효율이 높아진다. 이를 위해 APA(adaptive packetizing algorithm)를 사용하여 프레임의 크기를 수시로 조절하여야하며[2] 압축 과정에서는 압축률이 시간에따라 균일할수록 유리하게 된다. 비동기식 데이터의 전송에서도 압축률이 균일하지 않으면 완충기의 넘침이나 고갈을 일으키게 되고 이는 압축 과정의 중단이나 지연을 가져오게 되며 동기식 고속 데이터 전송에서는 이 현상이 더욱 심각해진다. 그러므로 고속으로 데이터를 전송하기 위한 압축 방법은 압축률이 높아야 할 뿐만아니라, 압축률이 균일하고, 배출 지연을 줄이기 위해서 빠른 압축 처리 속도가 필요하게 된다.

이 논문에서는 V.42bis를 기초로 한 고속 통신망에서 데이터 전송을 위한 알고리듬을 제안한다. 이를 위해, 부호책(dictionary) 크기에 의해 부호어의 길이가 결정 되므로, 부호책의 최적 크기를 실험을 통하여 분석하고 부호책을 더욱 빨리 적응적으로 변화시켜 압축률을 향상시켰으며, 부분 압축률의 변화를 조사 분석하고 압축 형식 전환을 단순하게 하여 기존의 방법보다 압축률을 고르게 하였다. 모의 실험으로 확인한 결과, 제안된 방법은 부분 압축률과 전체 압축률 면에서 기존의 방법보다 나은 성능을 보였고 따라서 데이터의 고속 전송에 보다 적합한 압축 방법이라고 생각할 수 있다. 나아가 동기식 압축방법을 제안하거나 표준화하는데에 이 논문에서 연구된 방법을 적용할 수 있을 것이다.

논문의 내용은 다음과 같다. 먼저, 2장에서는 V.42bis의 압축 방법을 소개하고 3장에서는 압축률을 향상시키고 압축률을 고르게 하기 위한 방법을 제안하며 4장에서는 제안 방법과 기존 방법의 성능을 비교하고 5장에서 결론을 맺는다.

Ⅱ. 데이터 압축 방법

일반적인 모뎀에 사용되는 비동기식 데이터 압축 방법은 전체적인 전송효율을 높이기 위해 높은 압축률 뿐만 아니라 빠른 처리 속도가 필요하다. 이를 만족시키는 대표적인 데이터 압축 방법이 ITU-T 표준 권고안인 V.42bis이다. 이 방법은 부호화 과정에서 주기적으로 압축률을 측정하여 문턱값(threshold)과 비교함으로써 압축 형식(compressed mode)과 비압축 형식(transparent mode)으로 나누어 전송한다. 비압축 형식에서는 입력 데이터를 압축 없이 그대로 전송하며 압축 형식에서는 LZW(Lempel-Ziv-Welch)방법을 변형한 형태를 사용하여 압축한 뒤 부호화하여 전송하게 된다.

압축 형식에서의 LZW 알고리듬은 부호화할 입력 문자(character)열을 부호책의 부호어들과 문자열 맞추기를 하여 일치하는 최대 길이로 문자열을 구문화

(parse)한 후, 구문화된 문자열에 해당하는 부호를 전송하는 방법이다[4]. 문자는 하나의 데이터 성분(single data element)을 말하며, 부호책에 없는 새로운 문자열들은 부호책에 추가되면서 부호책이 적응적으로 변하게 된다[5,6]. 부호책의 최대 크기는 고정되어 있으며 부호책 크기를 N이라고 할때 부호어를 이진(binary)수로 전송하는 경우 부호어의 길이는 $\lceil \log_2 N \rceil$($\lceil x \rceil$는 x보다 작지 않은 최소 정수)이 된다. 또한 부호책이 최대 크기를 넘어 갈 경우는 부호어 중에서 가장 사용빈도가 적은 것을 삭제하고 그자리에 새로 추가된 부호어를 넣는다. 전송된 부호어는 복호기측에서 부호화 과정과 같은 문자열 맞추기를 수행하고, 부호기측과 같은 부호책을 형성해가면서 복호화를 수행하게 된다. 예를 들어 부호어로 0과 1을 갖도록 부호책을 초기화하고, 이진 문자열 01011001을 부호화 한다면, 입력 문자열의 문자열 맞추기가 수행되면서 부호책에 01, 10, 011, 100이 등록되며, 입력 문자열은 0, 1, 01, 10, 01로 구문화되어 전송된다. 이 전송 과정을 정리하면 다음과 같다.

부호화 알고리듬

1. 입력데이터의 가능한 문자 모두를 각각 길이 1인 문자열로서 부호책에등록시켜 부호책을 초기화한다.
2. 부호책의 문자열과 입력 문자열을 비교하여 문자열 맞추기를 한다.
3. 주기적으로 검사하여 부분 압축률이 문턱값보다 작으면 비압축 형식으로,크면 압축 형식으로 전환한다.
4. 비압축 형식에서는 입력 문자 이진값을 그대로전송하고 압축 형식에서는 문자열 맞추기를 수행하여 부호책에서 찾은문자열의 위치를 나타내는 부호를 전송한다.
5. 맞추기가 이루어진 문자열에 (입력 데이터에서) 바로 다음 문자를 덧붙인 문자열을 부호책에 추가한다. (문자열 맞추기란 최대길이 문자열을 맞추는것이므로 추가되는 문자열은 부호책에 없는 문자열이다.)
6. 아직 부호화되지 않은 입력 문자열에 대해 과정 2부터 반복 수행한다.

복호화 알고리듬

1. 부호화 알고리듬에서와 같이 부호책을 초기화한다.
2. 비압축 형식에서는 전송된 데이터 그대로, 압축형식에서는 부호책에서 부호가 나타내는 위치의 문자열로 복호화한다.
3. 부호화 과정 5와 같은 방법으로 부호책에 문자열을 추가한다.
4. 새로 전송된 부호에 대해 과정 2부터 반복 수행한다.

Ⅲ. 제안 알고리듬

3.1 부호책 만들기

V.42bis의 압축 방법에 쓰이는 LZW 방법은 간단하며 높은 압축 효율을 보여 비교적 널리 사용되는 데이터 압축 방법이다. 그러나, 이 방법은 부호책이 어느 정도의 크기로 성장하기 전까지는 낮은 압축률을 보이며, 부호화 과정에서 새로운 구문에 다음 문자 하나만을 덧붙인 문자열을 부호책에 추가하고 구문화된 문자열 다음부터 새로운 문자열 맞추기를 시작하기 때문에 새로 만들어지는 구문과 직전에 만들어진 구문이 서로 단절되어 두 구문에 걸쳐 발생하는 문자열을 구문화할 가능성을 놓치는 단점이 있다. 이러한 단점을 개선하는 방법으로 구문화된 문자열에 다음 문자 둘 이상을 덧붙여 부호책에 추가함으로써 직전 구문과 이어지는 구문을 포괄하는 문자열을 부호책에 반영하고, 아울러 부호책 성장을 가속화시켜 압축률을 향상시키는 고속 부호책 갱신 알고리듬을 제안한다. 제안한 알고리듬은 기존의 방법과 달리 매 구문화시 추가되는 부호책의 문자열이 가변적이며, 기존의 방법이 놓칠 가능성이 있는 문자열을 부호책에 추가시키고 부호책을 가능한 한 빨리 성장시켜, 구문화 길이가 길어질 확률을 높임으로써 압축 성능 향상에 기여하게 된다. 구체적인 부호책 갱신 알고리듬은 3.3절에서 설명한다.

3.2 형식 전환 조절

V.42bis 방법에서는 주기적으로 압축률을 검사하여 문턱값보다 낮아지면 비압축 형식으로 문턱값보다 커지면 압축 형식으로 전환하여 전송을 하게 된다. 압축률을 검사하는 방법은 주기적으로 일정길이 데이터열의 압축되기 전의 비트수와 압축된 후의 비트수를 비

교하여 압축률을 결정한 뒤 문턱값과 비교하도록 되어 있으며, 일반적으로 압축률이 1보다 클 때에는 압축 형식으로 압축률이 1보다 작을 때에는 비압축 형식으로 전송을 하게 된다. 압축 방법은 부호책과의 문자열 맞추기를 이용한 방법이므로 입력 데이터가 지나간 데이터와 상관성을 갖지 않을 때에는 압축 효과가 낮아져 형식 전환을 일으키게 되며, 여기에 형식 전환을 나타내는 부가정보가 더해져 압축률을 더욱 저하시키게 된다. 즉, 일반적으로 압축 방법은 압축률이 균일하지 못하고 압축률이 균일하지 못한 경우에는 형식 전환이 빈번하게 일어나게 되며 결과적으로 압축률의 변화를 더욱 심하게 한다. 이러한 문제를 해결하려면 일시적인 압축률 저하에 의해 형식 전환이 일어나지 않도록 문턱값을 낮게 조절하여야 한다[7]. 문턱값을 1보다 낮게 하면 오히려 데이터의 비트수가 늘어나는 경우도 있지만 빈번한 형식 전환에 의해 더해지는 부가정보를 줄일 수 있게 되므로 이를 고려하여 문턱값을 결정하여야 한다. 실험적으로 여러 종류의 데이터에서 가장 좋은 성능을 보인 문턱값은 0.9였다.

3.3 알고리듬

제안된 알고리듬을 정리하면 다음과 같다.

부호화 알고리듬

1. 입력 데이터의 가능한 문자 모두를 각각 길이 1인 문자열로서부호책에 등록시켜 부호책을 초기화한다.
2. 부호책의 문자열과 입력 문자열을 비교하여 문자열 맞추기를 한다.
3. 부분 압축률이 문턱값보다 작으면 비압축 형식으로, 크면 압축 형식으로 전환한다. 이때 부분 압축률은 주기적으로 20개의 문자열마다 측정하며 문턱값을 0.9로 한다.
4. 비압축 형식에서는 입력 문자 이진값을 그대로 전송하고 압축 형식에서는 문자열 맞추기를 수행하여 부호책에서 찾은 문자열의 위치를 나타내는 부호를 전송한다.
5. 다음과 같은 부호책 갱신 알고리듬으로 부호책에 부호어를 추가한다.
   (1) 문자열 맞추기가 이루어진 문자열에 (입력 데이터에서) 바로 다음 문자를 덧붙인 문자열을 부호책에 추가시킨다.
   (2) 단계 (1)에서 부호책에 추가된 문자열에 (입력 데이터에서) 바로 다음 문자를 덧붙인 문자열을 부호책에 추가시킨다 : 부호책은 길이 1인 문자열로 초기화되어 길이 2,3,… 의 부호어가 차례로 추가되었으므로단계 (2)에서 추가하는 문자열은 부호책에 없다. 이것은 LZW 부호책의 공통된 성질이다.이 단계에서는 기존의 방법과 달리 2개의 문자가 추가된 문자열이 부호책에 추가될 수 있으며, 위의 과정을 반복하여 둘 이상의문자를 추가할 수도 있다.
6. 아직 부호화되지 않은 입력 문자열에 대해 과정 2부터 반복 수행한다.

복호화 알고리듬

1. 부호화 알고리듬에서와 같이 부호책을 초기화한다.
2. 비압축 형식에서는 전송된 데이터 그대로, 압축형식에서는 부호책에서 부호가 나타내는 위치의 문자열로 복호화한다.
3. 부호화 과정 5와 같은 방법으로 부호책에 문자열을 추가한다.
4. 새로 전송된 부호에 대해 과정 2부터 반복 수행한다.

Ⅳ. 모의실험 결과 및 고찰

기존의 알고리듬과 제안된 알고리듬의 성능을 비교하기 위해 세가지 다른종류의 데이터를 사용하여 실험하였다. 먼저, 데이터 압축기법의 성능을 결정하는 요인중의 하나인 부호책의 최대 크기 결정에 대한 분석 실험을 하였다. 이론적으로는 부호책이 무한히 클 때 가장 높은 압축률을 갖지만 실제적으로는 부호책의 크기에 비해 압축성능이 가장 좋게 되도록 그 크기를 결정하여야 한다.

실험 결과 부호책 크기가 512부터 4096까지일때는 압축률의 증가가 현격한 반면 그 이상에서는 압축 성능이 크게 변하지 않으므로 제한된 기억용량에서 가장 효율적인 성능을 보이는 크기가 4096임을 알수 있었다 (표 1, 그림 2).

표1. 부호책 크기와 압축률과의 관계

Tabel 1. Relation between the size of dictionary and thecompression ratio

Data type	512	1024	2048	4096	8196
C source code	1.11	1.46	1.81	2.06	2.13
paper	1.27	1.55	1.80	2.01	2.11
ps-file	1.03	1.32	1.46	1.52	1.57

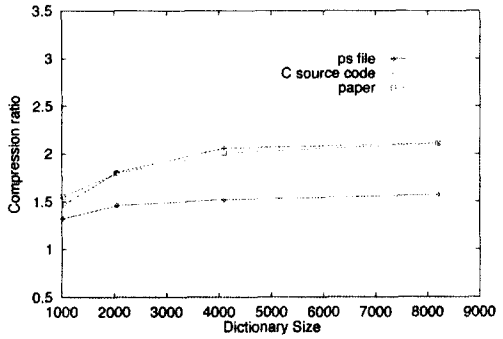


그림 2. 부호책 크기 결정을 위한 압축률 비교

Fig. 2. Compression ratio comparison for deciding thesize of dictionary

제안된 부호책 갱신 알고리듬(3.3절의 부호화 알고리듬 단계 5)에 의해 얻어진부호책을 사용하였을 때 제안 방법은 같은 부호책 크기와 형식 전환 방법을 사용한 기존의 방법에 비해 향상된 압축률을 보였다. 표 2는 제안된 방법과 기존 방법의 압축률을 비교한 것이다.

표 2. 부호책 갱신 방법에 대한 문자당 비트율

Tabel 2. Bit rates (per chatacter) for the twodictionary updating methods

Data type	기존방법	제안방법
C source code	3.52	3.30
paper	4.29	4.11
ps-file	4.29	3.28

다음으로 기존의 부호책 갱신 방법을 사용하고 부호화 과정의 형식 전환 문턱값만을 낮춤으로써 얻어진 실험 결과는 표 3, 4와 같다. 표 3은 문턱값을 1.0에서 0.9로 낮출 때 형식 전환 횟수가 반으로 줄어듬을 보여준다. 부분 압축률의 분산을 비교한 표 4는 제안 방법이 기존의 방법에비해 낮은 분산을 갖고 따라서 기존 방법에 비해 압축이 시간에 따라 균일하게되었음을 보여 준다.

표 3. 형식 전환 횟수

Tabel 3. Number of mode changes

Data type	기존방법	제안방법
C source code	1369	685
paper	853	427
ps-file	621	311

표 4. 형식 전환 방법에 대한 부분 압축률의 분산

Tabel 4. Variances of partial compressionratios for two mode transition methods

Data type	기존방법	제안방법
C source code	0.53	0.46
paper	0.37	0.32
ps-file	0.049	0.035

제안된 부호책 갱신 알고리듬을 사용하고 문턱값을 낮추면서 실행한 실험결과에서 문턱값 0.9가 1.0보다 전체 압축률(표 5)과 부분 압축률(그림 3,4,5)에서 나

표 5. 문턱값 1.0과 0.9에 대한 압축률

Tabel 5. Compression ratios for the thresholds 1.0 and 0.9

Data type	1.0	0.9
C source code	2.27	2.53
paper	1.86	2.07
ps-file	2.23	2.61

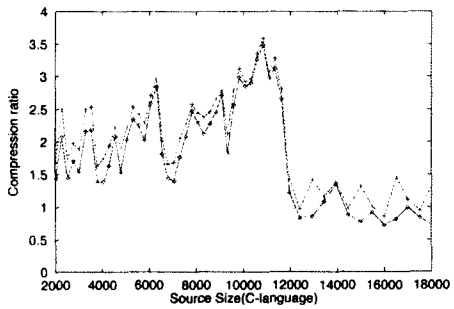


그림 3. 부분 압축률 비교 (C source code)실선:기존 알고리듬, 점선:제안 알고리듬

Fig. 3. Comparison of partial compression ratios(C source code)Full line : V.42bis, dot line : proposed algorithm

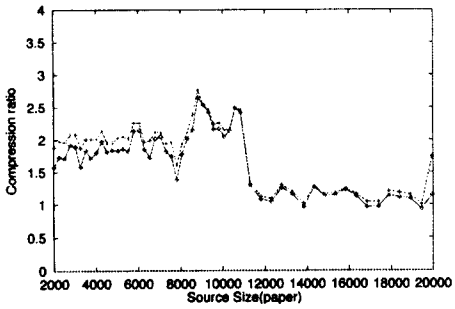


그림 4. 부분 압축률 비교 (paper)
실선 : 기존 알고리듬, 점선:제안 알고리듬

Fig. 4. Comparison of partial compression ratios(paper)
Full line : V.42bis, dot line : proposed algorithm

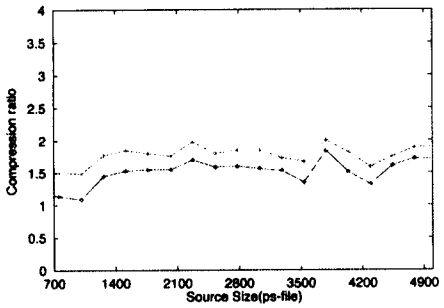


그림 5. 부분 압축률 비교 (ps-file)
실선:기존 알고리듬, 점선:제안 알고리듬

Fig. 5. Comparison of partial compression ratios(ps-file)
Full line : V.42bis, dot line : proposed algorithm

은 성능을 보여 데이터의 고속 전송에 보다 적합함을 확인하였다.

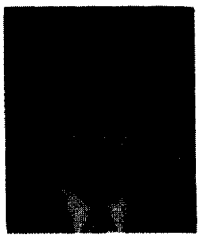
이 논문의 방법을 실제 적용하는 데에는 사실상 추가 부담이 없다. 추가되는 계산량은 전혀 없으며 부호책 갱신을 달리함으로써 부호책이 상대적으로 빨리 커진다는 차이가 있으나 부호책의 최대 크기가 정해져 있고 이보다 커질 수 없으므로 요구되는 메모리도 차이가 없다.

V. 결 론

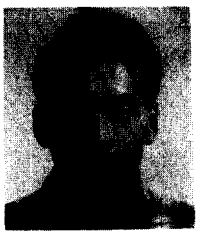
고정된 용량의 전송로에서 모뎀의 전송 처리량을 향상시키기 위해서비동기식 데이터 압축이 사용되고 있다. 일부 모뎀에서는 데이터 전송처리량을 늘리기 위해 동기식 데이터 압축방식을 사용하여 데이터를 전송하기도 한다. 이러한 경우 고속으로 데이터를 전송하기 위해서는 높은 압축률과 부분 압축률의 시간에 따른 변화가 적은 것이 요구된다. 기존의 통신망에서 사용되어온 V.42bis는 압축률이 균일하지 못하고 경우에 따라 압축 형식 전환이 지나치게 자주 일어나는 문제가 있어 고속 전송을 위한 데이터 압축에 부적합하다. 이 논문에서는 이러한 문제점을 해결하기 위해 압축률을 높이면서도 비교적 고른 압축률을 가질 수 있는 개선안을 제안하였다. 제안된 방법은 부호책의 갱신을 다양하게 하여 기존의 방법이 놓칠 가능성이 있는 문자열을 부호책에 추가시키고 부호책을 가능한 빨리 성장시켜 구문화된 문자열이 길어질 가능성을 높임으로써 결과적으로 기존의 방법에 비해 높은 압축률을 가지게 된다. 또한, 형식 전환 방법을 개선하여 부분 압축률의 변화가 적어지게 함으로써 압축률을 향상시키고데이터의 고속 전송에 적합함을 보였다.

참 고 문 헌

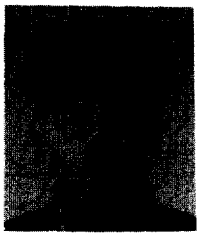
1. W. J. Betda, *Data Communication,*} Prentice Hall, 1996.
2. Motorola Information Systems Group, "A white paper on synchronous data compression over wide area networks," *Motorola, Inc.* May 8, 1996.
3. J. E. Mcnamara, *Technical Aspects of Data Communication*, Digital Press, 1988.
4. T. C. Bell, J. H. Cleary, I. H. Witten,*Text Comprssion, Prentice* Hall, 1990.
5. J. A. Storer, *Image and Text Comression*, Kluwer Academic Publishers, 1992.
6. J. Ziv, A. Lempel, "Compression of individual sequence via variable-rate coding," *IEEE Trans. on Information Theory,* vol. IT-24, pp. 530-536, 1978.
7. 조 성렬, ``고속 일반 데이터 전송을 위한 데이터 압축 방식의 연구," 석사 논문, 서울 대학교, 1997.



조 성 렬(Sung Yul Cho) 정회원
1968년 9월 17일생
1995년 2월 : 서울대학교 전자공학과(공학사)
1997년 2월 : 서울대학교 전자공학과(공학석사)
1997년 3월~현재:대우전자 디지털 TV연구소 연구원
※주관심분야 :동영상 부호화, 정보원 부호화 등임
csr@phoenix. dwe. co. kr



최 혁(Hyuk Choi) 정회원
1971년 2월 12일생
1994년 2월 : 서울대학교 전자공학과(공학사)
1996년 2월 : 서울대학교 전자공학과(공학석사)
1996년 3월~현재 : 서울대학교 전기공학부 박사과정
※주관심분야 :영상신호처리
camel@pine. snu. ac. kr



김 태 영(Tae Young Kim) 정회원
1974년 1월 23일생
1996년 2월 : 서울대학교 전기공학부(공학사)
1998년 2월 : 서울대학교 전기공학부(공학석사)
1998년 3월~현재 : 서울대학교 전기공학부 박사과정
※주관심분야 :신호처리
kty@pine. snu. ac. kr

김 태 정(Tae Jeong Kim) 정회원
통신학회 논문지 제21권 제6호 참조
pkim@snu. ac. kr