Compression of Multispectral Images

Enrico Piazza[†] and Heung-Kook Choi[††]

ABSTRACT

This paper is an overview of research contributions by the authors to the use of compression techniques to handle high resolution, multi-spectral images. Originally developed in the remote sensing context, the same techniques are here applied to food and medical images. The objective is to point out the potential of this kind of processing in different contexts such as remote sensing, food monitoring, and medical imaging and to stimulate new research exploitations. Compression is based on the simple assumption that it is possible to find out a relationship between pixels close one each other. In multi-spectral images it translates to the possibility to say that there is a certain degree of correlation within pixels belonging to the same band in a close neighbourhood. Once found a correlation based on certain coefficient on one band, the coefficients of this relationship are, in turn, quite probably, similar to the ones calculated in one of the other bands. Based upon this second observation, an algorithm was developed, able to reduce the number of bit/pixel from 16 to 4 in satellite remote sensed multispectral images. A comparison is carried out between different methods about their speed and compression ratio. As reference it was taken the behaviour of three common algorithms, LZW (Lempel-Ziv-Welch), Huffman and RLE (Run Length Encoding), as they are used in common graphic format such as GIF, JPEG and PCX. The presented methods have similar results in both speed and compression ratio to the commonly used programs and are to be preferred when the decompression must be carried out on line, inside a main program or when there is the need of a custom made compression algorithm.

멀티 스펙트럴 영상들의 압축

Enrico Piazza[†] · 최흥국[††]

요 약

본 논문은 고해상도에 대한 멀티 스펙트럴 영상들에 대한 압축기술에 대한 연구입니다. 원래는 원격센싱 컨텍스트에 대한 개발로 이를 식품과 의료영상에 적용하였습니다마는 이러한 가능성을 여러 컨텍스트에서 처리하는 것을 목표로 두었으며 즉, 원격센싱, 식품모니터링 그리고 의료영상의 새로운 분야로 탐구 및 적용하였다. 압축은 한 화소와 관계한 이웃 간의 화소들 간의 간단한 추정에 기반하여 나타날 수 있도록 하였다. 멀티 스펙트럴 영상들은 화소들이 같은 밴드 안에서 가까이 이웃하여 있는 어떤 상접한 정도의 관계를 해석하였으며 하나의 발견된 상관관계는 어떠한 한 밴드내에서의 계수에 기반 한다. 그 계수와의 관계는 다른 밴드에서 계산되어진 것과 유사하다. 두 번째의 관찰에서는 개발되어진 알고리즘이 화소당 비트수를 멀티 스펙트럴 위성원격영상에서 16비트에서 4비트로 감소 할 수 있었다. 따라서 다른 방법론들과 속도 및 압축률에 대해서 비교 분석하였다. 보통 그래픽 포맷인 GIF, JPEG 그리고 PCX를 사용하였으며 참조와 같이 LZW, Huffman과 RLE의 알고리즘을 행하였다. 소개되어진 방법들은 압축을 줄이는 것이 선상, 프로그램 안에서 혹은 관례적인 압축 알고리즘에서 속도와 압축률에서 유사한 결과를 가져왔다.

Key words: Multi-spectral images, Speed and compression ratio, Compression algorithm.

접수일 : 2002년 3월 5일, 완료일 : 2002년 9월 27일
본 연구는 보건복지부 보건의료기술진흥사업의 지원에 의하여 이루어진 것임 (02-PJ1-PG3-20601-0003).
† Park Air Systems, Horten, Norway
†† 종신회원, 인제대학교 컴퓨터공학부 조교수

1. Introduction

A 24 bits full-colour image contains definitively more information than the 8 bits or 12 bits monochrome image[1-3]. Therefore, the trend of image acquisition in several fields like, for instance, microscopy, is changing from grey-level to colour image. When handling, transmitting, storing and processing multi-spectral images, especially remote sensed, given the enormous quantity of data to deal with (just think that a single passage of the one radiometer flown aboard a polar satellite often exceeds 100 Mbytes) it appears convenient to use techniques of compression of data without loss of information[4]. Originally developed in the remote sensing context, the same techniques are here applied to food images. As known the most efficient compression techniques must be chosen taking into account the particular class of images being manipulated. In this case appears convenient to take advantage of the correlation that exists between the various bands of one multi-spectral image.

There are several compression programs available, both commercial and on the public domain, with different performances in terms of compression ratio and time employed in their execution. However, often, it is useful to have a simple and robust algorithm that, sacrificing its performances, is easy to be embedded inside custom software or directly within the programs for processing the images. This is the reason for which it was developed a program that based upon the local correlation between adjacent pixels, through a differences algorithm, succeeds to compress the images by a factor in excess of two with acceptable times[5].

Even[6] presents a few compression schemes, *including the* Lempel-Ziv solution. Piazza[7] presents a remote sensed Multi-spectral Images compression algorithm. This work mainly deals with the compression of remote sensed Images gathered from a radiometer flown aboard an environmental satellite. In this work some NOAA / AVHRR[8] images have been used with focus on the southern Europe and the Mediterranean Basin. Larmore[9] introduces an algorithm for generating length-limited Huffman codes, to be used in compressing files. Limiting the codes to a known length is useful because you can then pack the codes into known size words. For instance, as given here, you can guarantee codes <= 16 bits, which means the codes can be stored in two-byte words. Without such length limiting, there's no guarantee that the codes would fit. The algorithm used is called the Package-Merge algorithm.

2. Multi-spectral images

Multi-spectral images are images in which every picture element (pixel) is described by means of several parameters or spectra. The several parameters can be the three primary colours of the visible spectrum or the several channels of a remote sensing radiometer. A multi-spectral image can have from one to several tens of components. At the same time, a three components image in which every component is coded with 8 bits, can be regarded as a mono-spectral image with a depth of 24 bits per pixel. On the other side, a grey scale image, coded with 8 bits a pixel, can be regarded as a multi-spectral image with 8 components, each

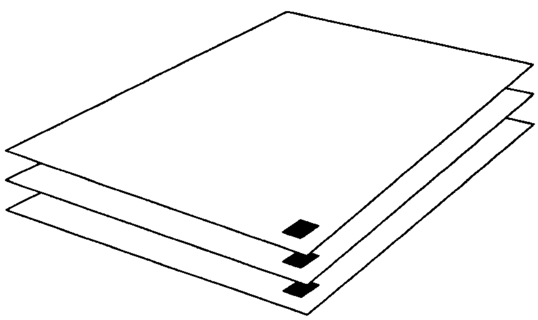

Fig. 1. The image viewed as three planes. Each plane is an image itself representing only one of the components.

coded with 1 bit. In this case, every component assumes the name of "bit-plane".

For example, satellites such as NOAA [8] polar, Meteosat, DMSP (Defence Meteorological Satellite Program), Landsat, and others have passive sensors that detect and measure energy reflected and/or emitted from the earth's surface in different wavelengths of the electromagnetic spectrum. The portions of the detected spectrum by each satellite's sensors are designated by bandwidths, or bands for short, and each band is assigned a numerical identifier. Different spectral bands are used to gain information about various materials on the earth's surface such as water, vegetation and soil. Because each material uniquely absorbs and reflects energy throughout the spectrum, these materials produce their own distinct spectral responses or signatures.

3. Images used to test the presented algorithms

The Images used to test the algorithms presented in this work comes from three different fields of the imaging technology: satellite remote sensing, food inspection and microscopy.

The satellite remote sensed images have been extracted from the 5 bands of a full resolution digital image sensed by the AVHRR radiometer flown on board the NOAA-14 polar satellite. NOAA-16 replaced NOAA-14 on March 20, 2001. It was launched September 21, 2000. NOAA-14 was launched in December 1994. They represent a portion of the Mediterranean Sea in the visible (bands 1-2 of AVHRR) and IR bands (band 3-5 of AVHRR). All the images, for each band, have dimensions 1306×1319×16 bits (of which 10 bits/pixel informative). The images have 800 bytes of header thus they are 3,446,028 bytes in size. The images were received at the Electronic Engineering Department of the University of Firenze, where, in the PIN-Centro Studi Ingegneria, Prato facility, is operative a primary receiving station for both NOAA HRPT and Meteosat PDUS Images. Fig. 2 shows one example of a remote sensed image used for the test. This class of images is called R in the result section.

The food inspection images have been extracted from colour images of beef M. Longissimus dorsi captured by a Sony DCS-D700 camera. The same exposure and focal distance were used for all images. Digital colour photography was carried out with a Hama repro equipment (Germany).

Green colour was used as background and photographs were taken on one side of the meat.

The meat pieces were enlightened with two lamps, with two fluorescent tubes each (15 W). Polaroid filters were used on the lamps and on the camera. Images are 1344×1024 pixels with a resolution of 0.13×0.13 mm. The original image has a dimension of 4 128 768 bytes. These images were courtesy of Lucia Ballerini at the Centre for Image Analysis, Uppsala, Sweden. Fig. 3 shows the image used for the test called "M" in the result section.

The microscopy images were obtained courtesy of the School of Information and Computer Engineering, Inje University, Gim-Hae, Korea. Fig. 4 and Fig. 5 show the P1 and P8 cellular images used for the test respectively.

Fig. 2. One example of a remote sensed image used for the R test. 1306×1319 pixels, 5 bands (R=2, G=3 and B=4 in the printed image).

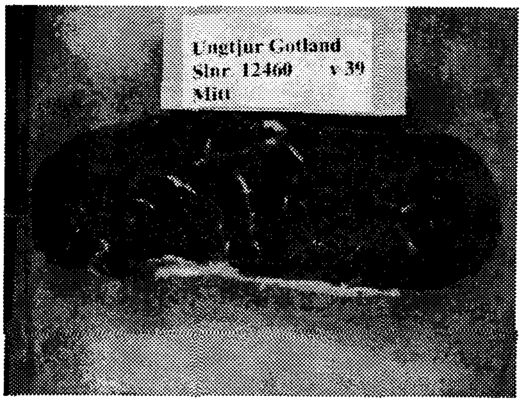

Fig. 3. The "M" image used for the test.1344×1024 pixels, 3 bands (R, G and B).

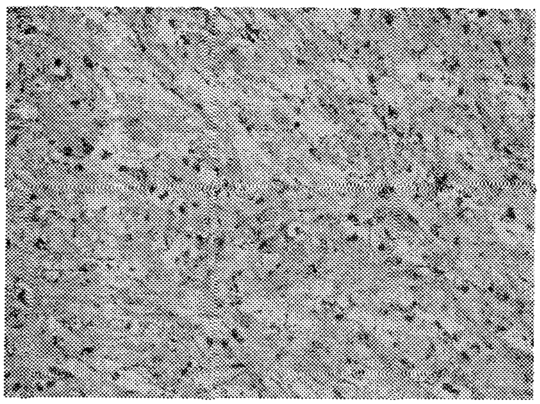

Fig. 4. The "P1" cellular image used for the test. 2048×1536 pixels, 3 bands (R, G and B), file p1010046.tif.

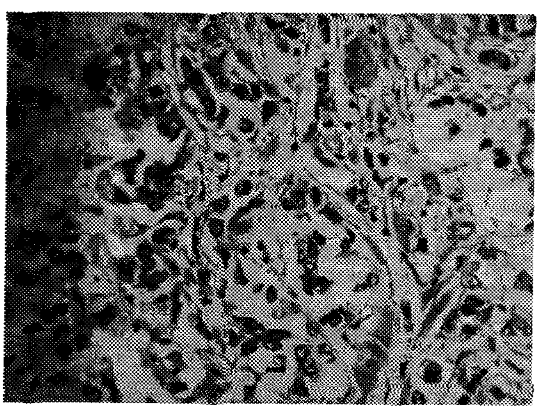

Fig. 5. The "P8" cellular image used for the test. 2048×1536 pixels, 3 bands (R, G and B), file p8060001.tif.

4. Images compression algorithms

In this section are presented the different compression schemes tested:

- LZW as used in GIF, even if it is the base for other compression software like PKZIP and LHA.
- Huffman as used in the JPEG standard.
- RLE as used in the PCX standard
- A method developed by the authors.

Each one of the well known algorithms, is introduced by a short summary of pros & cons and a discussion of the major issues. The authors method is presented in more details.

4.1. LZW (GIF)

Pros:

- Standard
- Good Compression;

Cons:

- Need to build the Colormap
- Medium Easy to implement
- Only 256 colours
- Loss of information due to the limited number of colours
- Copyright and Patent problems
- Max three components;

'GIF' (tm) is CompuServe's standard for defining generalized colour raster images. This 'Graphics Interchange Format' (tm) allows high-quality, high-resolution graphics to be displayed on a variety of graphics hardware and is intended as an exchange and display mechanism for graphics images. GIF images are compressed to reduce the file size. The technique used to compress the image data is called LZW (after LempelZiv-Welch) and was first described by Welch[10]. Welch work was based on the work of Ziv and Lempel [11,12].

The compression is based upon recurring strings in the input stream in such a way that frequent long strings are stored with a single code word. The code is built dynamically while coding, and is not passed to the decoder that builds the same code while decoding. The Variable-Length-Code LZW Compression is a variation of the Lempel-Ziv Compression algorithm in which variable-length codes are used to replace patterns detected in the original data. The algorithm uses a code or translation table constructed from the patterns encountered in the original data; each new pattern is entered into the table and its index is used to replace it in the compressed stream. The compressor takes the data from the input stream and builds a code or translation table with the patterns as it encounters them; each new pattern is entered into the code table and its index is added to the output stream; when a pattern is encountered which had been detected since the last code table refresh, its index from the code table is put on the output stream, thus achieving the data compression.

Unisys holds a patent on the procedure described in the article by Welch[10]. While having the right to pursue legal action or seek damages against infringing LZW developers and publishers, Unisys has so far been very accommodating and fair. GIF files are not covered of the patent. There is no risk in distributing GIF files or in using the GIF name. Only the software employing the LZW algorithm for writing GIF files is "at risk". The Unisys patent includes claims that specifically cover the decompression of LZW-compressed material, so it may also affect simple GIF readers. Unisys does not require licensing, or fees to be paid, for non-commercial, non-profit GIF-based applications, including those for use on the online services. No license or license fees are required for non-commercial, not-for-profit GIF-based applications or for non-commercial, not-for-profit GIF-freeware, so long as the LZW capability provided is only for GIF.

The LZW algorithm used in GIF matches algorithmically with the standard LZW algorithm with the following differences:

1. A special Clear code is defined which resets all compression/decompression parameters and tables to a start-up state.

2. An End of Information code is defined that explicitly indicates the end of the image data stream. LZW processing terminates when this code is encountered. It must be the last code output by the encoder for an image.

3. The output codes are of variable length, up to 12 bits per code. This defines a maximum code value of 4095. Whenever the LZW code value would exceed the current code length, the code length is increased by one. The packing/unpacking of these codes must then be altered to reflect the new code length.

Because the LZW compression used for GIF creates a series of variable length codes, of between 3 and 12 bits each, these codes must be reformed into a series of 8-bits bytes that will be the characters actually stored or transmitted. This provides additional compression of the image.

4.2 Huffman (Jpeg)

Pros:

- Standard
- Cheap
- Good Compression
- No Copyright or Patent problems (Public domain under JPEG general license)
- 24 bits images;

Cons:

- Heavy loss of information
- Difficult to implement
- Max three components;

JPEG (pronounced "jay-peg") is a standardized compression method for full-colour and grey-scale images. JPEG is intended for "real-world" scenes; cartoons and other non-realistic images are not its strong suit. JPEG is loss, meaning that the output image is not necessarily identical to the input image. Hence you should not use JPEG if you have to have identical output bit. However, on typical images of real-world scenes, very good compression levels can be obtained with no visible change, and amazingly high compression levels can be obtained if you can tolerate a low-quality image. A format such as JPEG cannot be used as a substitute for GIF. Unlike GIF, JPEG was designed as a "loss" format. This means that it slightly changes an image as it is compressed. This is unacceptable for many applications. Also, while JPEG excels in compressing real world true colour images, it offers no support for palette-based images.

The compression is based upon decimating the coefficients of the 2D Discrete Cosine Transform of the input image. The file format is called JFIF. This format has been agreed to by a number of major commercial JPEG vendors, and is the de facto standard. For more detail about the JPEG standard see the work of Wallace[13]. It appears that the arithmetic coding option of the JPEG spec is covered by patents owned by IBM and AT&T, as well as a pending Japanese patent of Mitsubishi. Hence arithmetic coding cannot legally be used without obtaining one or more licenses. For this reason, support for arithmetic coding has been removed from the free JPEG software. Since arithmetic coding provides only a marginal gain over the un-patented Huffman mode, it is unlikely that very many people will choose to use it.

4.3 RLE (PCX)

Pros:

- Standard
- Fast
- Easy to implement
- 24 bits images;

Cons:

- Medium Compression
- Copyright and Patent problems
- Max three components;

The image file consists of encoded graphic data. The encoding method is a simple byte oriented run-length technique. When more than one colour plane is stored in the file, each line of the image is stored by colour plane (generally ordered red, green, blue, intensity), As shown below:

Scan line 0:	RRR..	(Plane 0)
	GGG...	(Plane 1)
	BBB...	(Plane 2)
	III...	(Plane 3)
Scan line 1:	RRR..	(Plane 0)
	GGG...	(Plane 1)
	BBB...	(Plane 2)
	III...	(Plane 3)

Basically, the scheme compares adjacent bytes in one scan-line. If the adjacent bytes are the same, it the writes out a flag (0×C0) with a count (actually 0×C0 | count). The next byte is the data byte that is repeated. For example, if five bytes are the same 01 the PCX output is only two bytes C5 01, where C5 means special flag (C) with a count of 5, and the 01 in the next position is the data byte. If there is no match in the next byte, PCX files just use the plain data byte (implicit count of 1). For example, if the byte stream is 01 01 01 01 01 04 01 01 the PCX output is C5 01 04 C2 01.

This PCX format breaks down in two cases: i) If the byte value is >= 0×C0 then the format uses a C1 and ii) If there are more than 15 bytes the same, the count makes the 0×C0 become 0×D0 or higher. PCX only uses counts under 15.

4.4 Piazza 98

Pros:

- Fast
- Easy to implement
- Published Work
- Good compression;

Cons:

- No Standard;

A remote sensed multi-spectral images compression algorithm was first presented in 1998[14]. This work mainly deal with the compression of remote sensed images gathered from a radiometer flown aboard an environmental satellite. In this work some NOAA / AVHRR image has been used with focus on the southern Europe and the Mediterranean Basin. The deal is to find a compression algorithm to reduce the memory requirement for the recording of remote sensed images. In order to store m bits/pixel (8, 4 or 2 bits/pixel) images out of the M bit/pixel. That is 8 bits/pixel for digital camera images or 10 bits/pixel like in the case of the NOAA-AVHRR. It is possible to find out a relationship between pixels close one each other in such a way it is possible to say that there is a certain degree of correlation within pixels belonging to the same band in a close neighbourhood.

The single band algorithm works line by line as in the follow:

Given

b1(t); $0 \le t \le N-1$

in where

t pixel index

b1(t) value of t-th pixel of band 1

N Number of pixels to process

let compute

b1c(t); $0 \le t \le N-1$

that is the band 1 compressed.

For each row:

The first element of the compressed row is the full resolution value of the uncompressed one.

b1c(0) = b1(0); // stores M bit

old = b1(0)

then compute the difference between the current pixel on the row and the old value

d(t) = b1(t) - old.

If the value of d(t) is in the range $2^{(m-1)}-1$, $2^{(m-1)}-1$ (-127, 127 if m = 8 bits), then this value is stored and the old value is replaced with the current value:

b1c(t) = d(t); // stores m bit

old = b1(t);

else it is stored the key and the full resolution, M bit, value. As a key it is used the value $-2^{(m-1)}$ (-128 if m = 8) and the old value is not updated in order to keep the last known "good" value.

b1c(t) = key:b1(t); // stores m+M bit

Table 1. Range and key given the length of the code m.

m	range	key
2	-1, 1	-2
4	-7, 7	-8
8	-127, 127	-128

The multiband algorithm acts as in the following.

Given

b1(t); $0 \le t \le N-1$

b2(t); $0 \le t \le N-1$

where

t pixel index

b1(t) value of t-th pixel of band 1

b2(t) value of t-th pixel of band 2

N Number of pixels to process

let compute

b1c(t); $0 \le t \le N-1$

that is the band 1 compressed on the basis of band 2.

For each t we find the parabola that pass through b2(t-1), b2(t) and b2(t+1).

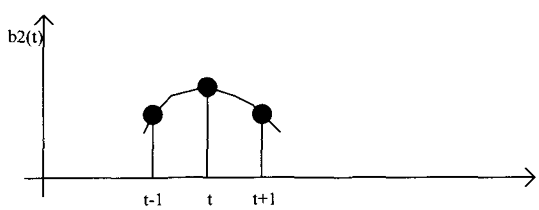

Fig. 6. The estimation of a parabolic law on the basis of the band b2.

Placing the zero of the x axis in the t position we obtain:

$$b2(t) = a_{02} + a_{12}\Delta t + a_{22}\Delta t^2$$
$$a_{02} = b2(t)$$

being $\Delta t = 1$, then the expression became:

$$b2(t) = a_{02} + a_{12} + a_{22}$$

Solving the system, it is possible to explicitly write the expressions of a_{12} and a_{22}.

$$a_{12} = \frac{b2(t+1) + b2(t-1)}{2} - b2(t)$$
$$a_{22} = \frac{b2(t+1) - b2(t-1)}{2}$$

then an estimate of b1(t+1) is computed by applying the same parabola to the just find coefficients:

$$a_{01} = b1(t)$$
$$b12(t) = a_{01} + a_{12}1 + a_{22}1^2$$

and the error is finally computed:

$d12(t) = |b12(t) - b1(t)|$

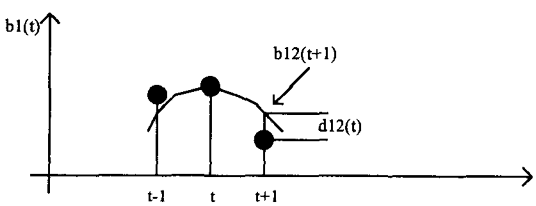

Fig. 7. The estimation of the value of the band B1 on the basis of a parabolic law found on the band b2.

If the value of d12(t) is in the range $2^{(m-1)}-1$, $2^{(m-1)}-1$ (-127, 127 if m = 8 bits), then this value is stored, else it is stored the key and the M bit value. As a key it is used again the value $-2^{(m-1)}$ (-128 if m = 8)

4.5 How much image information is lost when using the JPEG compression

This is a very difficult question. Initially the Shannon Entropy was considered to measure the information carried by the images. It turned out that, sometimes, the original image had a lower entropy of the one compressed by JPEG. The entropy is a measure of how many bit are necessary to code the image. It is expected to be less than the true color 24 bit used to store it.

p8060001, entropy=14.156394, jpg entropy= 14.081643
p8060002, entropy=14.395470, jpg entropy= 14.193002
p1010046, entropy=12.864973, jpg entropy= 12.917777
s12460m, entropy=13.054569, jpg entropy= 12.807890

It looks like the compression actually adds some information. In some way it is true. The JPEG compression acts on the image adding some noise, which as a first approximation can be considered white with entropy of 7 to 9 bit.

p8060001, dif entropy=7.921017
p8060002, dif entropy=9.384736
p1010046, dif entropy=7.524231
s12460m, dif entropy=7.236326

5. Results

The presented methods have been tested on an Intel Pentium MMX 200 MHz based computer where they took only few seconds to carry on the task to compress the whole image. As an example of result, in the following Table 2 there are reported the size of the compressed file and the time employed by the single band algorithms. Alongside the graphic formats introduced in the previous section, three compression software, easily available on the network, have been tested. PKZIP and LHA use the Lempel-Ziv-Welch (LZW) technique.

The HZIP program and associated files implements an algorithm for generating length-limited Huffman codes, to be used in compressing files. Limiting the codes to a known length is useful because you can then pack the codes into known size words. For instance, as given here, you can guarantee codes <= 16 bits, which means the codes

Table 2. Remote sensed images. Dimension of compressed files and the time in seconds the algorithm took to their compression.

FILE	COMPR	DIMENSION[Byte]	RATIO [%]	TIME	
				[S]	[s/MB]
ORIGIN	—	17,230,140	1.00	0.0	0.00
LHA	LZW	7,713,191	0.448	103.3	6.29
PKZIP	LZW	7,466,468	0.433	55.3	3.36
HZIP	HUFFMAN	10,946,207	0.64	47.0	2.86
MCOMPRA	PIA/2	5,692,005	0.33	37.5	2.28

Table 3. Food images. Dimension of compressed files and the time in seconds the algorithm took to their compression.

FILE	COMPR	DIMENSION[BYTE]	RATIO [%]	TIME	
				[S]	[s/MB]
ORIGIN	—	4,303,324	1.000	0.0	0.00
LHA	LZW	2,444,523	0.568	25.8	6.29
PKZIP	LZW	2,301,058	0.535	10.8	2.63
HZIP	HUFFMAN	3,942,383	0.916	10.4	2.53
GIF	LZW8bits	793,659	0.184	50.2	12.23
JPEG	HUFFMAN	122,100	0.028	39.4	9.60
PCX	RLE/24	3,890,841	0.904	18.4	4.48
COMPR4	PIA/4	2,187,836	0.508	6.7	1.63
COMPR2	PIA/2	1,485,924	0.345	5.9	1.44

can be stored in two-byte words. Without such length limiting, there's no guarantee that the codes would fit. The algorithm used is called the Package-Merge [9]. A version originally written by Bryan Flamig, Azarona Software, Denver, CO. was used.

The algorithms were applied to a remote sensing set of image [14] (5 bands) each of 3446028 bytes. These bands are 10 bit deep and are not suitable for compression with standard image compression algorithms. The results obtained with the remote sensed images were promising, see Table 2, so other classes of images were tested.

Table 4. Cellular microscopy images. Dimension of compressed files and the time in seconds the algorithm took to their compression.

FILE	COMPR	DIMENSION[BYTE]	RATIO [%]	TIME	
				[S]	[s/MB]
ORIGIN	—	879,883	1.000	0.0	0.00
LHA	LZW	389,037	0.442	5.2	6.19
PKZIP	LZW	375,782	0.427	4.8	5.72
HZIP	HUFFMAN	492,125	0.559	1.3	1.54
GIF	LZW8bits	294,046	0.334	5.9	7.03
JPEG	HUFFMAN	78,170	0.089	1.1	1.31
PCX	RLE	965,977	1.097	1.9	2.26
MCOMPRA	PIA/2	829,071	0.942	2.6	3.10

The algorithms were applied to a food quality set of image [5] (3 bands: R, G and B). The results obtained were, again, very promising, see Table 3. The same algorithms were then applied to a set of cell microscopy images [1] each of 879,883 bytes and gave slightly worse results. The following Table 4 shows these results obtained.

Table 5. Cellular microscopy images. Dimension of compressed files and the time in seconds the algorithm took to their compression (File p1010046).

FILE	COMPR	DIMENSION[BYTE]	RATIO [%]	TIME	
				[S]	[s/MB]
ORIGIN	—	9,451,593	1.000	0	0.000
LHA	LZW	6,706,665	0.710	516	57.246
PKZIP	LZW	6,437,062	0.681	233	25.849
HZIP	HUFFMAN	7,676,269	0.812	172	19.082
GIF	LZW8bits	2,665,142	0.282	701	77.770
JPEG	HUFFMAN	467,338	0.049	718	79.656
PCX	RLE	10,075,166	1.066	280	31.064
MCOMPR4C	PIA/4	5,743,429	0.608	142	15.754
MCOMPR2C	PIA/2	5,567,910	0.589	177	19.637

Sometimes it happens that being the dynamic range of the image large enough, the 2 bits compression actually result in bigger files than the 4 or 8 bits one. The same algorithm was applied to another couple set of cell microscopy images [1] each of 9,451,593 bytes and gave just about the same results of the meat and remote sensed images results.

The following table shows these results obtained. More cellular images were then tested, with a larger size, to see if it was that particular image or the whole set of images, which led to a worse performance of the algorithm.

Table 6. Cellular microscopy images. Dimension of compressed files and the time in seconds the algorithm took to their compression (File p8060001).

FILE	COMPR	DIMENSION[BYTE]	RATIO [%]	TIME [S]	TIME [s/MB]
ORIGIN	—	9,451,593	1.000	0.0	0.000
LHA	LZW	7,209,857	0.763	401	44.488
PKZIP	LZW	7,020,793	0.743	258	28.623
HUFF-ZIP	HUFFMAN	8,234,745	0.871	175	19.415
GIF	LZW	2,685,082	0.284	742	82.319
JPEG	HUFFMAN	396,215	0.042	952	105.617
PCX	RLE	9,126,884	0.966	292	32.395
MCOMPR4C	PIA/4	5,482,878	0.580	136	15.088
MCOMPR2C	PIA/2	5,134,279	0.543	164	18.194

In the following Fig. 8 there is a comparison between the various compression methods by plotting the results on a size / time chart. The points in the lower-left corner are the best results, meaning a small size obtained in a little time. The points in the upper-right corner, are the worse results, meaning a large size obtained in a longer time.

The key to read the figure is: M: meat image; R: remote sensed image; C: Cellular microscopy image, P1: cellular microscopy file p1010046; P8: cellular microscopy file p8060001.

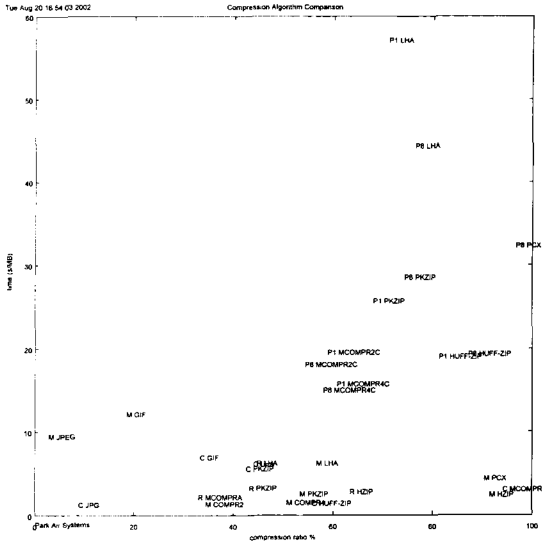

Fig. 8. Compression algorithm comparison. M: meat image; R: remote sensed image; C: Cellular microscopy image, P1: cellular microscopy file p1010046; P8: cellular microscopy file p8060001. JPG and GIF offer always the best compression ratio at the price of loss of data and slower algorithms. Our method offers about the same compression ratio of LHA or PKZIP but with a somewhat faster algorithm.

As it is possible to see, the bigger the file, the slower the algorithm. This is a general rule valid for all the tested methods. Moreover, it is possible to see that the algorithm has a similar behaviour in relation one other. JPG and GIF offer always the best compression ratio at the price of loss of data and slower algorithms. LHA is always in the top right corner offering slow algorithm and bigger compressed files, together with PCX and with approximately the same compression ratio of PKZIP but with a slower algorithm. Our method offers about the same compression ratio but with a somewhat faster algorithm.

The presented methods have similar results in both speed and compression ratio to the commonly used programs and are to be preferred when the decompression must be carried out on line, inside a main program or when there is the need of a custom made compression algorithm.

6. Acknowledgment

The Graphics Interchange Format (c) is the Copyright property of Compuserve Incorporated. GIF(sm) is a Service Mark property of Compuserve Incorporated. The HZIP software is COPYRIGHT (c) 1991 AZARONA SOFTWARE. The food images are courtesy of Lucia Ballerini at the Centre for Image Analysis, Uppsala, Sweden. The authors wish to thank the PIN-Centro Studi Ingegneria, Prato, Italy, that run, in connection with the Electronic Engineering Dept. of the University of Florence a primary receiving station for AVHRR images.

The PCX, JPEG and GIF compression have been carried out by the means of the convert software included in the package ImageMagick 5.2.3 00/09/01 Copyright (C) 2000 ImageMagick Studio. The LHA compression have been carried out by the means of the program LHA version 2.13 Copyright (c) Haruyasu Yoshizaki, 1988-91. The PKZIP compression have been carried out by the means of the program

PKZIP (R) FAST! Create/Update Utility, Version 2.04g, 02-01-93 Copyright. 1989-1993 PKWARE Inc. All Rights Reserved. Shareware Version

References

[1] Choi Heung-Kook, Lee Byeong-Il, Choi Hyun-Ju, "A Segmentation Method Using ML Classification for Breast Carcinoma Images", *The fifth Korea-Germany joint workshop on Advanced Medical Image Processing*, Seoul, Korea, May 15-16, 2001.

[2] Fairchild M.D., *Color appearance models*, Addison Wesley Longman inc., pp. 173-230, 1998.

[3] Wyszeki G., Stiles W.S., *Color science: concepts and methods, quantitative data and formulae*, John Wiley, pp. 63-243, 1992.

[4] Rao A.K., Bhargava S., "Multispectral Data Compression Using Bidirectional Interband Prediction", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 34, No. 2, pp. 385, March 1996.

[5] Piazza Enrico, "Food Multispectral Image Compression", *The fifth Korea-Germany joint workshop on Advanced Medical Image Processing, Seoul, Korea*, May 15-16, 2001.

[6] Even Shimon, "Four value-adding algorithms", *IEEE Spectrum*, May 1998.

[7] Piazza E., *Integrazione di dati multispettrali da satelliti meteorologici differenti per il telerilevamento dell'ambiente marino*, PhD Thesis, University of Florence, Florence, Italy, June 1998 (in Italian).

[8] Noaa, *Polar Orbiters Data User Guide*, NOAA Report, Boulder, Colorado, 1996.

[9] Larmore Lawrence L., Hirschberg Daniel S., "A Fast Algorithm for Optimal Length-Limited Huffman Codes", *JACM*, Vol. 37, No.3, pp. 464-473, July 1990.

[10] Welch, Terry A., "A technique for high-performance data compression", *IEEE Computer*, pp. 8-19, June 1984.

[11] Ziv Jacob, Lempel Abraham, "A universal algorithm for sequential data compression", *IEEE Transactions on Information Theory*, pp. 337-343, May 1977.

[12] Ziv Jacob, Lempel Abraham, "Compression of individual sequences via variable-rate coding", *IEEE Transactions on Information Theory*, pp. 530-536, September 1978.

[13] Wallace Gregory K., "The JPEG Still Picture Compression Standard", *Communications of the ACM*, Vol. 34, No. 4, pp. 30-44, April 1991.

[14] Piazza Enrico, "Remote Sensed Multispectral Images Compression", *SPIE 98 Conference*, San Diego, CA, July 1998.

Enrico Piazza

Enrico Piazza received a degree in Electronic Engineering from the University of Florence, Italy in 1993. He holds a Ph.D. in Methods and Technologies for Environmental Monitoring. He is presently with Park Air Systems, Norway, working on developing algorithms for the NOVA 9000 Advanced Surface Movement Guidance and Control System in the European Commission "VISION" and "BETA" research and development projects.

최 흥 국

1988년 Linköping University, Computer Engineering, Linköping, Sweden(공학사)
1990년 Linköping Universiy, Computer Engineering, Linköping, Sweden(공학석사)
1996년 Uppsala University, Computer Image Analysis, Uppsala, Sweden(공학박사)
1996년~1997년 서울대학교 의공학연구소
1997년~현재 인제대학교 정보컴퓨터공학부 조교수
1997년~1999년 인제대학교 전산정보처장
1997년~현재 한국멀티미디어학회 및 신호처리 · 시스템학회 국제상임이사
관심분야 : 멀티미디어, 컴퓨터그래픽스, 영상처리 및 분석

교 신 저 자

최 흥 국 (621-749) 경남 김해시 어방동 607 인제대학교 컴퓨터공학부