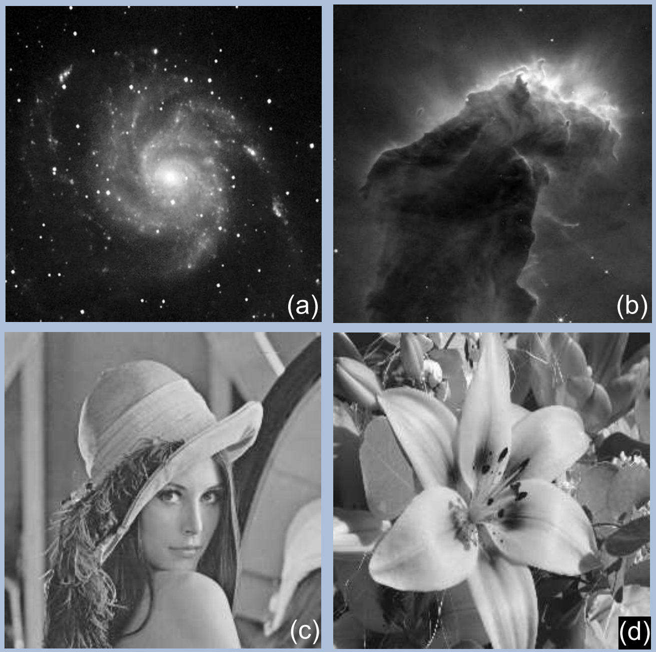
Holographic Data Storage in Photoactive Media

Ravinder Kumar Banyal
and
B. Raghavendra Prasad
Indian Institute of Astrophysics
Kormangala, Bangalore - 560034
E-mail: brp@iiap.res.in



Outline

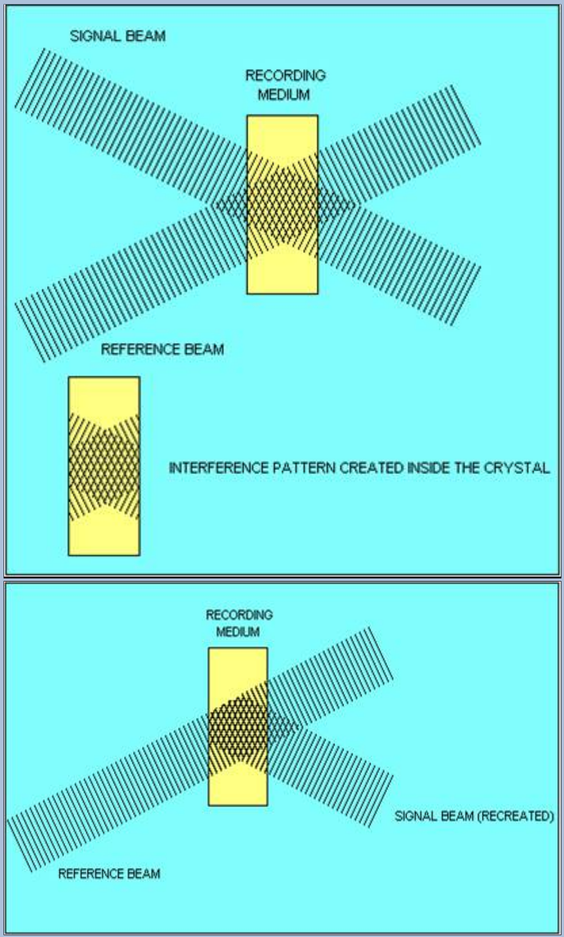
IMAGE COMPRESSION: –Haar transformed based image compression, methods and implementation in HDSS.

HOLOGRAPHIC DATA STORAGE: –Components of HDSS, binary and astronomical image storage in Fe:Ce:Ti $LiNbO_3$ crystal, experimental results.

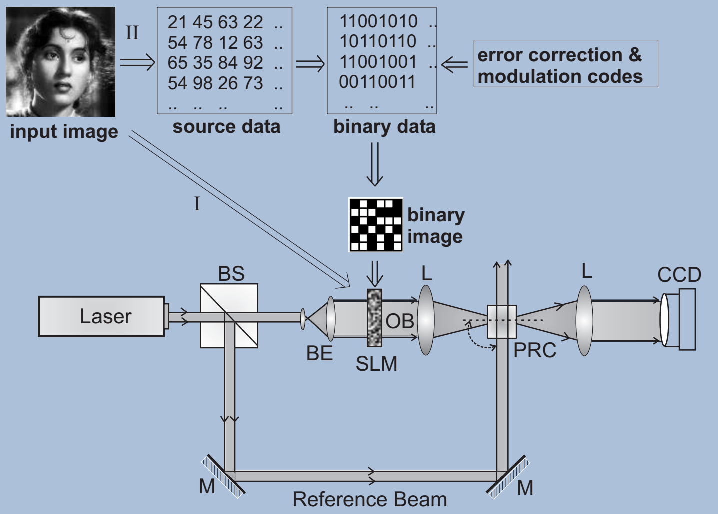
Introduction

- Massive amount of data is produced from weather forecasting, multimedia applications, remote sensing, medical imaging, astronomical data from ground or spaced based observation.
- Semiconductor based 2D storage devices (hard disks, magnetic tape drives, CDs, DVDs etc.) are likely to reach their fundamental limit in near future.
- Potential alternative for future storage applications
  - volume (3D) holographic data storage using PR media
- Salient features
  - extremely high data storage density $> 1\text{Tbit/cm}^3$
  - high speed read/write operation $> 1\text{Gbit/s}$
  - rapid access time $\approx 10\mu\text{s}$–1ms
  - immunity against noise
  - associative retrieval & parallel search capabilities

Principle of holographic data storage



Holographic Data Storage and image compression



A general layout of holographic data storage system (HDSS).

⇒ Two possible schemas for HDSS

- direct storage of grey scale images
- binary storage
  - ∗ implementation of powerful error correction & modulation codes.
  - ∗ necessary to preserve quantitative information in an image.

Data Compression

- Data is the means by which information is conveyed. Data compression refers to reducing the amount of data required to represent a given quantity of information or event.
- Varying amount of data may represent same amount of information. The presence of unimportant data that does not provide any new or extra information leads to data redundancy.
- Types of data redundancies in digital images:
  - Interpixel redundancy
  - Coding redundancy
  - Psychovisual redundancy
- The data compression is achieved by reducing or eliminating one or more of these redundancies.

† Each piece of information or event is represented by a sequence of symbol (letter, number, bit etc.) called code word. For example, ASCII code for letter 'A' and digit '9' is 65 and 57, respectively. Corresponding 8-bit binary representation is: 01000001 and 00111001.

Interpixel redundancy –Arises from the correlation among the pixels due to structural or geometrical similarities between the objects in the image. Examples are: transform based coding, run-length coding, LZW coding, bit-plane coding.

Coding redundancy – The gray levels are encoded in such a way that uses more code symbol than absolutely necessary to represent each gray level. The coding redundancy can be overcome by *variable-length* coding, e.g., Huffman coding, arithmetic coding.

Psychovisual redundancy – Refers to information that are redundant for visual perception of the image. Example: 16-bit and 24-bit images are indistinguishable for human eyes. Removing this redundancy may result in loss of quantitative information from the image.

Wavelet Transform

Mathematical transformations are applied to a signal to obtain information that are not present in raw signal, e.g.,

spatial-domain representation $\overset{\mathcal{F.T.}}{\Longleftrightarrow}$ frequency-domain representation An

analogy:
Fourier transform is like a prism that decomposes a white light into its constituent colours (spectral decomposition.)

A wavelet transform is similar to a microscope that shows the details of an object at different scales (multiscale analysis.)

Haar transform is commonly employed for image compression and can be expressed in matrix form as

$$T = W^T \cdot P \cdot W$$

where, P, W and T are $N \times N$ image, transformation and Haar transformed matrices, respectively. The transformation matrix W contains the Haar basis functions.

How the wavelet transform works?

- The Haar transform exploits the statistical or interpixel redundancy present in the natural images to achieve data compression.
- It uses a method of *averaging and differencing* for manipulating the matrices.

y	128	120	124	132	124	120	112	116
y_1	124	128	122	114	4	-4	2	-2
y_2	126	118	-2	4	4	-4	2	-2
y_3	122	4	-2	4	4	-4	2	-2

Red = *Approximation coefficients* $\left(\frac{\text{sum of pair}}{2}\right)$
Blue = *Detail coefficients* $\left(\frac{\text{difference of pair}}{2}\right)$

- The process of *averaging and differencing* is reversible.
- For an array of length 2^n, n steps are needed to carry out the above operation.

Using matrix algebra

$$y_1 = (124\ 128\ 122\ \ 114\ 4\ -4\ \ 2\ -2) = (128\ 120\ 124\ 132\ 124\ 120\ 112\ 116)A_1$$
$$y_2 = (126\ 118\ -2\ \ \ 4\ \ 4\ -4\ \ 2\ \ -2) = (124\ 128\ 122\ 114\ \ 4\ -4\ \ 2\ \ -2)A_2$$
$$y_3 = (122\ \ \ 4\ -2\ \ \ 4\ \ 4\ \ -4\ \ 2\ \ -2) = (126\ 118\ \ -2\ \ \ 4\ \ 4\ -4\ \ 2\ \ -2)A_3$$

$$A_1 = \begin{pmatrix} 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 \end{pmatrix};$$

$$A_2 = \begin{pmatrix} 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ and}$$

$$A_3 = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

$\Longrightarrow$ This set of operations can all be done in one step

$$y_3 = y \cdot W$$

$\Longrightarrow$ where W is transformation matrix

$$W = A_1 \cdot A_2 \cdot A_3$$

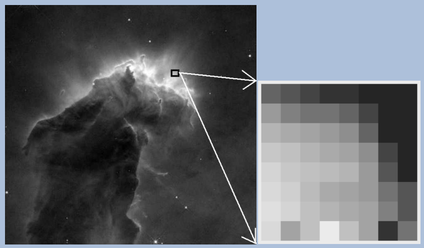
$\Longrightarrow$ Also note that each column of A_i matrices that comprise W is orthogonal to every other. The matrices are invertible. Thus:

$$W^{-1} = A_3^{-1} \cdot A_2^{-1} \cdot A_1^{-1}$$

$\Longrightarrow$ This means we can get our original data back using

$$y = y_3 \cdot W^{-1}$$

An example



A 8×8 pixel block from Eagle Nebula in M16

$$P = \begin{pmatrix} 115 & 114 & 113 & 112 & 112 & 110 & 106 & 103 \\ 119 & 117 & 116 & 116 & 115 & 113 & 109 & 105 \\ 122 & 121 & 120 & 119 & 118 & 115 & 111 & 108 \\ 125 & 124 & 122 & 121 & 120 & 118 & 113 & 110 \\ 127 & 125 & 124 & 123 & 122 & 119 & 114 & 111 \\ 128 & 126 & 123 & 121 & 120 & 119 & 116 & 114 \\ 129 & 127 & 124 & 122 & 121 & 120 & 117 & 114 \\ 128 & 120 & 124 & 132 & 124 & 120 & 112 & 116 \end{pmatrix}$$

Note: The last row is our vector y.

First we want to average and difference the rows of matrix P. In order to get the row averaged matrix Q we multiply P on right by transformation matrix W.

That is $Q = P \cdot W$, therefore

$$Q = \begin{pmatrix} 110.62 & 2.87 & 1.00 & 3.25 & 0.50 & 0.50 & 1.00 & 1.50 \\ 113.75 & 3.25 & 1.00 & 3.50 & 1.00 & 0.00 & 1.00 & 2.00 \\ 116.75 & 3.75 & 1.00 & 3.50 & 0.50 & 0.50 & 1.50 & 1.50 \\ 119.12 & 3.87 & 1.50 & 3.75 & 0.50 & 0.50 & 1.00 & 1.50 \\ 120.62 & 4.12 & 1.25 & 4.00 & 1.00 & 0.50 & 1.50 & 1.50 \\ 120.87 & 3.62 & 2.50 & 2.25 & 1.00 & 1.00 & 0.50 & 1.00 \\ 121.75 & 3.75 & 2.50 & 2.50 & 1.00 & 1.00 & 0.50 & 1.50 \\ 122.00 & 4.00 & -2.00 & 4.00 & 4.00 & -4.00 & 2.00 & -2.00 \end{pmatrix}$$

We can again *average and difference* the columns of above matrix by left multiplying Q with W^T. Therefore, the haar transformed matrix T becomes:

$$T = W^T \cdot P \cdot W = \begin{pmatrix} 118.19 & 3.66 & 1.09 & 3.34 & 1.19 & 0.00 & 1.12 & 1.06 \\ -3.12 & -0.22 & 0.03 & 0.16 & -0.56 & 0.37 & 0.00 & 0.56 \\ -2.87 & -0.37 & -0.12 & -0.12 & 0.12 & -0.12 & -0.12 & 0.12 \\ -0.56 & 0.00 & 0.81 & -0.06 & -0.75 & 1.12 & -0.12 & 0.75 \\ -1.56 & -0.19 & 0.00 & -0.12 & -0.25 & 0.25 & 0.00 & -0.25 \\ -1.19 & -0.06 & -0.25 & -0.12 & 0.00 & 0.00 & 0.25 & 0.00 \\ -0.12 & 0.25 & -0.62 & 0.87 & 0.00 & -0.25 & 0.50 & 0.25 \\ -0.12 & -0.12 & -2.25 & -0.75 & -1.50 & 2.50 & -0.75 & 1.75 \end{pmatrix}$$

—Correspondence between P and T.

Threshold and Rounding off

Select a threshold value ϵ and replace any elements of T less than ϵ to ZERO.
This will result in a sparse matrix S. If we choose $|\epsilon| = 2$ then

$$S = \begin{pmatrix} 118 & 4 & 0 & 3 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 3 & 0 & 0 \end{pmatrix}$$

S has only 7 nonzero coefficients compared to 64 elements in the original matrix P.
Therefore, it becomes much easier to store or transmit the sparse matrices.

— But, 'how well' we can reconstruct P from S?

Loss–less and Lossy Compression

$$\epsilon = \begin{cases} 0 & \text{then } S = T \text{ which is called loss–less compression,} \\ \geq 0 & \text{some of the elements of } T \text{ are reset to zero} \\ & \text{which leads to image distortion or lossy compression.} \end{cases}$$

The degree of compression is measured by compression ratio:

$r = \frac{\#\text{ non–zero entries in transformed matrix } T}{\#\text{ non–zero entries in } S}$; $55/7 \approx 7.8$ in the above case.

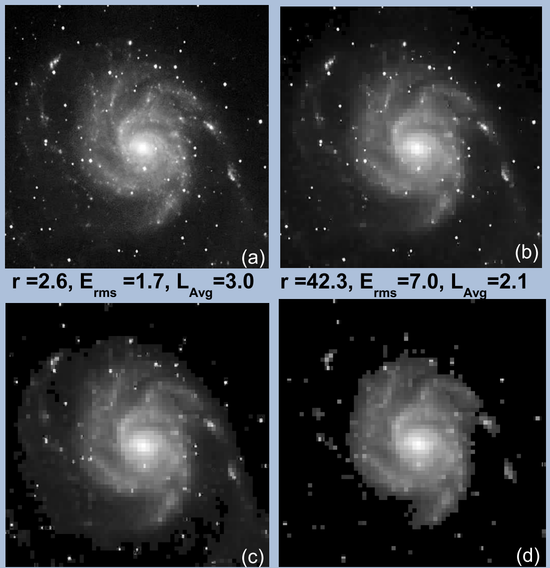
The mean square error (MSE) is

$\text{MSE} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [P(i,j) - P'(i,j)]^2$; –which is 4.78 in the above case.

Some Examples

Uncompressed Test Images

M101 Galaxy



r =2.6, E_{rms} =1.7, L_{Avg} =3.0 r =42.3, E_{rms} =7.0, L_{Avg} =2.1

r =92, E_{rms} =14.4, L_{Avg} =2.1 r =241, E_{rm} =25.5, L_{Avg} =2.0

M101 Galaxy

The effect of threshold ε on r, $E_{\rm rms}$, $L_{\rm Avg}$ (bits/pixel) and the visual image quality of four representative images. (a) $\varepsilon = 1$, (b) $\varepsilon = 10$, (c) $\varepsilon = 25$ and (d) $\varepsilon = 50$.

Leena Image

Lina Image

The effect of threshold ε on r, E_{rms}, L_{Avg} (bits/pixel) and the visual image quality of four representative images. (a) $\varepsilon = 1$, (b) $\varepsilon = 10$, (c) $\varepsilon = 25$ and (d) $\varepsilon = 50$.

Binary Encoding

⇒ At basic level the storage and transmission of data take place in binary form.

⇒ Therefore, prior to storage or transmission, the data is coded to streams of 1's and 0's.

⇒ A fixed length binary coding of source symbols results in coding redundancy.

e.g.,

Symbol(G.L.)	8-bit binary code
11	00001011
246	11110110
125	01111101
31	00011111

red: redundant bits

⇒ The coding redundancy can be lifted by variable length coding of the source symbols.

Examples: Huffman and arithmetic coding.

Huffman Coding

- Huffman coding yields the smallest possible number of codes per source symbol.
- Most frequently occurring symbols (pixel values) are assigned lesser number of bits.

Original Source		Source reduction			
Symbol (gray level value)	Probability (p_i)	1	2	3	4
x_3	0.43	0.43	0.43	0.43	0.57
x_1	0.21	0.21	0.23	0.34	0.43
x_2	0.12	0.13	0.21	0.23	
x_5	0.11	0.12	0.13		
x_6	0.08	0.11			
x_4	0.05				

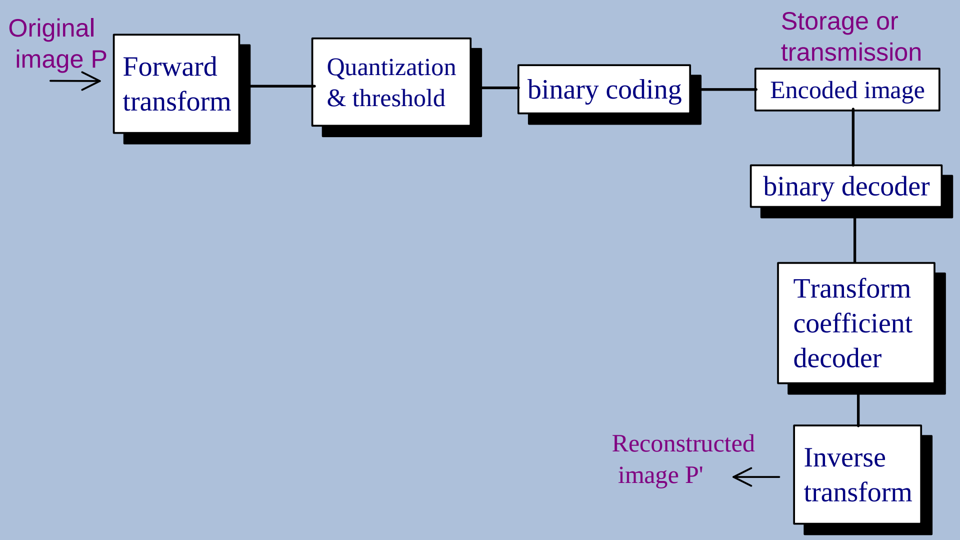
Original Source			Source reduction			
Symbol	Prob. (p_i)	Code	1	2	3	4
x_3	0.43	**1**	0.43 1	0.43 1	0.43 1	0.57 0
x_1	0.21	**010**	0.21 010	0.23 00	0.34 01	0.43 1
x_2	0.12	**001**	0.13 011	0.21 010	0.23 00	
x_5	0.11	**000**	0.12 001	0.13 011		
x_6	0.08	**0110**	0.11 000			
x_4	0.05	**0111**				

- Unique Prefix Property: no code is a prefix to any other code, so decoding is trivial as long as the coding table is available.

Example

Binary coded data stream: 01110100011001101 · · Decoded symbols: $x_4, x_1, x_6, x_6, x_3 \cdot\cdot$

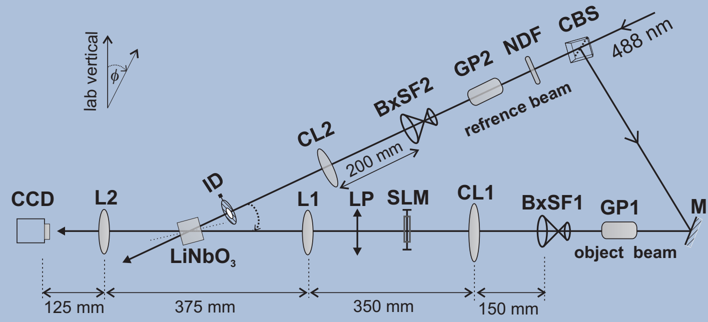
$$L_{\text{avg}} = \sum_{i=1}^{6} p_i(x_i)\ l_i(x_i) = 0.43\times1 + 0.21\times3 + 0.12\times3 + 0.11\times3 + 0.08\times4 + 0.05\times4 = 2.27 \text{ bits/pixel}$$



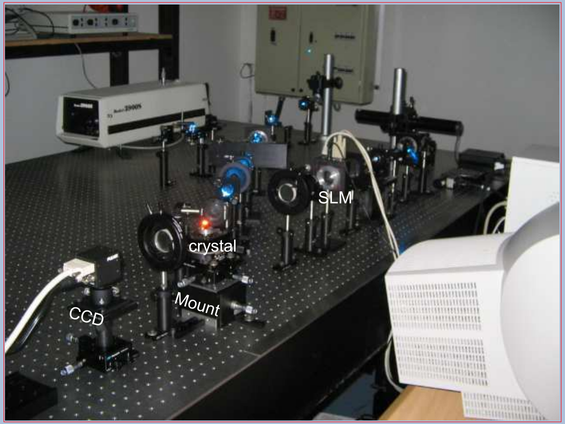
Summary of image compression scheme for HDSS.

Storage of Astronomical & binary images

Experimental Scheme



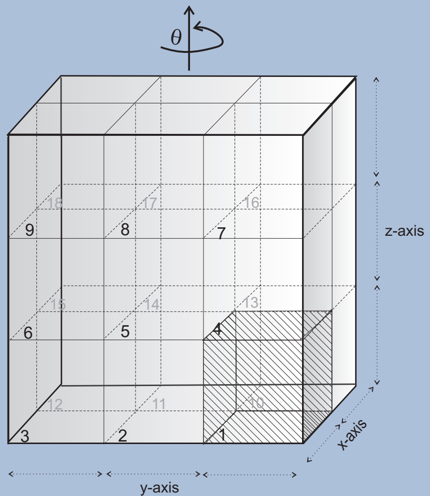
Fourier image plane geometry for HDSS in the lab.



A photograph of holographic data storage and retrieval system in the lab.

Storage of Binary & Astronomical images

Spatial and Rotational Multiplexing

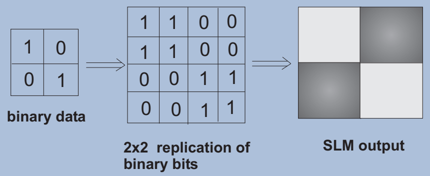


Partition of the crystal (Fe:Ce:Ti doped $LiNbO_3$) to implement spatial and rotational multiplexing.

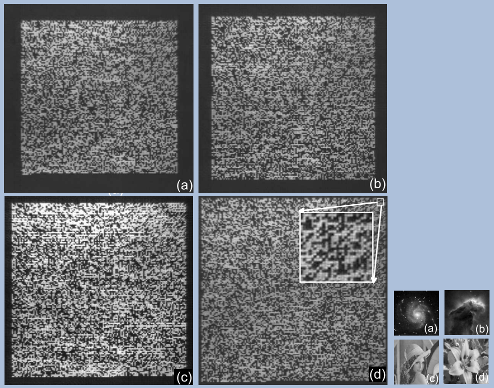
Addressing scheme: (R, θ)
R: spatial location (1–18) & θ: rotational position during hologram recording (1–6°).

Storage of Binary & Astronomical images

- The image data was Haar transformed as explained earlier.
- For loss-less compression, the threshold ε was set to zero.
- The negative coefficients in Haar transformed data were up-shifted (made positive) by adding a number $c_{\min}$ to each coefficient, where $c_{\min}$ was largest negative coefficient in the Haar transformed data.
- *Up-shifted* Haar coefficients were replaced by corresponding minimum length binary codes obtained using Huffman encoding method. The resulting 1-dimensional bit stream had N elements.
- Huffman encoded streams of 1s and 0s was partitioned into two-dimensional square page of size $l \times l$. Here, $l = \sqrt{N}$. For l to be an integer, N needs to be padded with additional 0s.
- Each data bit on SLM was represented by 2×2 replication of original binary bits as shown in Figure. SLM pixels were transparent for white blocks representing 1s and opaque for greyed-out blocks representing 0s.



Storage of Binary & Astronomical images



Retrieved binary pages of (a) M101 galaxy, (b) Eagle nebula, (c) Lena and (d) Lily. The reconstructed (64×64 pixel size) images after the binary decoding are shown in the right.

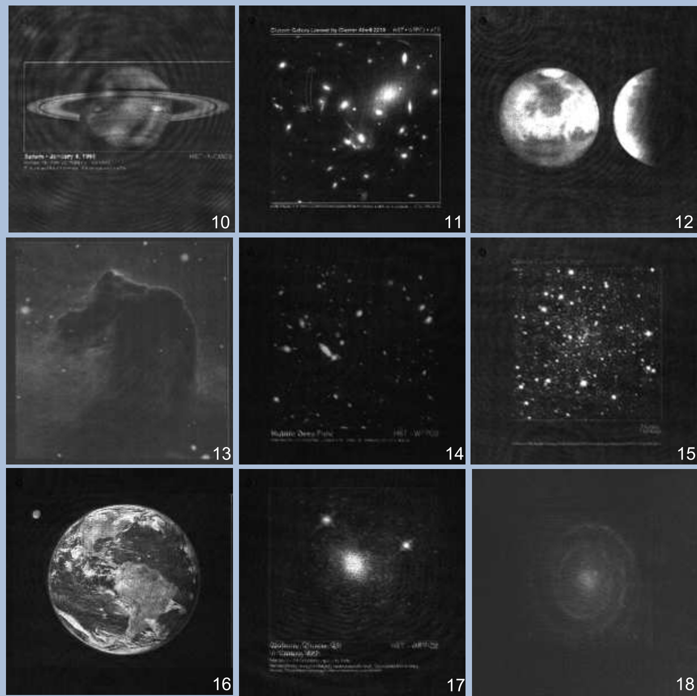
The main steps involved in reconstruction are as follow:

- ▷ The retrieved data pages were block-processed using a programming code written in *Mathematica* and Huffman encoded sequence of 1s and 0s was reconstructed.
- ▷ Using Huffman decoding algorithm, the binary sequence was re-mapped to obtain *up-shifted* Haar transformed data.
- ▷ c_{min} was subtracted from the up-shifted coefficients to unfold the negative Haar coefficients.
- ▷ Subsequently, inverse Haar transformation was applied to recover the original images.

Storage of Astronomical Images in Fe:Ce:Ti:$LiNbO_3$

- A holographic data-base of 106 astronomical images was created in 18 spatial locations of the crystals.
- Each spatial location contains a stack of 6 holograms recorded by rotating the crystal about an axis perpendicular to the plane of recording beams.
- The minimum angular separation between subsequent holograms was$\Delta\theta \approx 1^\circ$.
- Recording exposure time for each image was around 4-5 minutes.
- During the read and write operations, the laser input power was fixed at 0.5 W.
- Every third and sixth raw image retrieved from each of the 18 hologram stacks is shown next.

Storage of Astronomical Images in Fe:Ce:Ti:$LiNbO_3$



Holographically retrieved images from the crystal.
The set comprises every third and sixth raw image
read-out from all the 18 spatial locations of the crystal.

S. No	Image description	Original size & Source
1	Gravitational lensing in galaxy cluster Abell 2218	400×250 [a]
2	Artist's impression of planets' motion	350×431 [b]
3	Eagle nebula	300×335 [a]
4	Star formation regions in Tarantula nebula	500×375 [b]
5	The Sombrero galaxy M104 NGC 4594	431×348 [b]
6	Spiral galaxy NGC 4414	431×350 [c]
7	Spiral galaxy pair NGC 3314	400×500 [a]
8	Buzz Aldrin on the Moon	242×298 [b]
9	Galaxy NGC 6781	407×528 [c]
10	Saturn image	500×267 [a]
11	Distance galaxy lensed by Abell 2218	610×655 [a]
12	Earth-Mars photograph taken from space	350×250 [b]
13	Horse head nebula	400×446 [a]
14	Hubble deep field	400×446 [a]
15	Globular cluster NGC 3697	300×375 [a]
16	Earth from the outer space	431×348 [c]
17	Globular cluster in galaxy M31	400×446 [a]
18	Backwards spiral Galaxy NCG 4622	431×350 [c]
19	Spiral galaxy NGC 7331	734×587 [c]
20	Galaxy M106	660×547 [c]
21	Kashima radio antenna	540×420 [–]
22	High energy gamma ray telescope at IAO*	640×480 [d]
23	Galaxy M71	458×366 [b]
24	Galaxy M81	998×713 [c]
25	Artist's conception of black hole & a companion star	350×280 [a]
26	Jupiter comet impact	300×200 [a]
27	Galaxy NGC 1637	404×378 [d]
28	Galaxy NGC 6946	721×611 [c]
29	Himalayan Chandra telescope at IAO	400×300 [d]
30	Galaxy M86	760×510 [c]
31	Saturn image	256×256 [–]
32	Planetary nebula M2-9	400×262 [a]
33	Galaxy M101	750×542 [a]
34	Jupiter from Cassini orbiter	431×348 [b]
35	Whirlpool galaxy M51	431×350 [b]

[a] http://hubblesite.org/
[b] http://www.astroimages.net/
[c] http://www.gralak.com/Astro.html/
[d] http://www.iiap.res.in/
* IAO is Indian Astronomical

Observatory located at Hanle (J&K).

Storage of Astronomical Images in $Fe:Ce:Ti:LiNbO_3$

Few note worthy observations:

- Due to uniform exposure time, the fainter images are recorded poorly as compared to brighter ones (e.g. image 9, 13, 18, 30 etc.)
  - This problem can be eliminated by selecting a *weighted exposure* time schedule that takes into account the energy content (image brightness) in the image.
- A typical astronomical image have very large dynamic range. A nonlinear response of the crystal may cause over saturation of the brighter objects in the image before dim objects are recorded. As a consequence, the brighter features in the retrieved image, appear bleached out or over-exposed (e.g., image 12, 25 and 27).
  - An appropriate gamma correction of the input image prior to recording would help curtailing the selective saturation.
- The quality of raw images retrieved from the crystal can be improved by various digital image processing means. For example, the interference pattern present in image 4, 10 and 28 can be removed by band-pass filtering in Fourier domain. Likewise, the pixel-to-pixel random noise can also be minimized using local averaging or Gaussian filtering of the image pixels in spatial domain.

Questions??