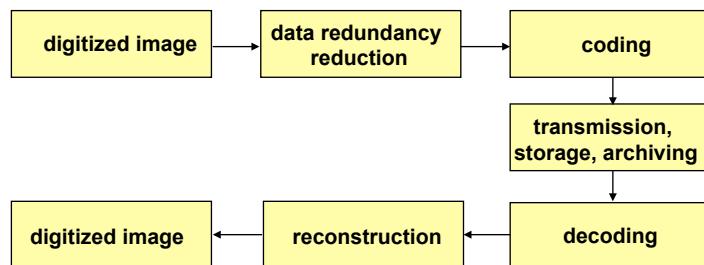


## Image Data Compression

Image data compression is important for

- image archiving e.g. satellite data
- image transmission e.g. web data
- multimedia applications e.g. desk-top editing

Image data compression exploits redundancy for more efficient coding:



1

## Run Length Coding

Images with repeating greylevels along rows (or columns) can be compressed by storing "runs" of identical greylevels in the format:

greyvalue1 repetition1 greyvalue2 repetition2 ...

For B/W images (e.g. fax data) another run length code is used:

row # column # run1 begin column # run1 end column # run2 begin column # run2 end ...

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0															
1															
2															
3															

run length coding:  
(0 3 5 9 9)  
(1 1 7 9 9)  
(3 4 4 6 6 8 8 10 10 12 14)

2

1

## Probabilistic Data Compression

A discrete image encodes information redundantly if

1. the greylevels of individual pixels are not equally probable
2. the greylevels of neighbouring pixels are correlated

Information Theory provides limits for minimal encoding of probabilistic information sources.

Redundancy of the encoding of individual pixels with G greylevels each:

$$r = b - H$$

b = number of bits used for each pixel  
=  $\lceil \log_2 G \rceil$

$$H = \sum_{g=0}^{G-1} P(g) \log_2 \frac{1}{P(g)}$$

H = entropy of pixel source  
= mean number of bits required to encode information of this source

The entropy of a pixel source with equally probable greylevels is equal to the number of bits required for coding.

3

## Huffman Coding

The Huffman coding scheme provides a variable-length code with minimal average code-word length, i.e. least possible redundancy, for a discrete message source. (Here messages are greylevels)

1. Sort messages along increasing probabilities such that  $g^{(1)}$  and  $g^{(2)}$  are the least probable messages
2. Assign 1 to code word of  $g^{(1)}$  and 0 to codeword of  $g^{(2)}$
3. Merge  $g^{(1)}$  and  $g^{(2)}$  by adding their probabilities
4. Repeat steps 1 - 4 until a single message is left.

Example:

message	probability	code word	coding tree	Entropy: H = 2.185	Average code word length of Huffman code: 2.2
$g^{(5)}$	0.3	00	0		
$g^{(4)}$	0.25	01	1	0.55	
$g^{(3)}$	0.25	10	0		
$g^{(2)}$	0.10	110	0	0.45	
$g^{(1)}$	0.10	111	1	0.20	

4

2

## Statistical Dependence

An image may be modelled as a set of statistically dependent random variables with a multivariate distribution  $p(x_1, x_2, \dots, x_N) = p(\underline{x})$ .

Often the exact distribution is unknown and only correlations can be (approximately) determined.

Correlation of two variables:

$$E[x_i x_j] = c_{ij}$$

Covariance of two variables:

$$E[(x_i - m_i)(x_j - m_j)] = v_{ij} \text{ with } m_k = \text{mean of } x_k$$

Correlation matrix:

$$E[\underline{x} \underline{x}^T] = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \dots \\ c_{21} & c_{22} & c_{23} & \dots \\ c_{31} & c_{32} & c_{33} & \dots \end{bmatrix}$$

Covariance matrix:

$$E[(\underline{x} - \underline{m})(\underline{x} - \underline{m})^T] = \begin{bmatrix} v_{11} & v_{12} & v_{13} & \dots \\ v_{21} & v_{22} & v_{23} & \dots \\ v_{31} & v_{32} & v_{33} & \dots \end{bmatrix}$$

Uncorrelated variables need not be statistically independent:

$$E[x_i x_j] = 0 \quad \cancel{\rightarrow} \quad p(x_i x_j) = p(x_i) p(x_j)$$

For Gaussian random variables, uncorrelatedness implies statistical independence.

5

## Karhunen-Loève Transform

(also known as *Hotelling Transform* or *Principal Components Transform* )

Determine uncorrelated variables  $\underline{y}$  from correlated variables  $\underline{x}$  by a linear transformation.

$$\underline{y} = A(\underline{x} - \underline{m})$$

$$E[\underline{y} \underline{y}^T] = A E[(\underline{x} - \underline{m})(\underline{x} - \underline{m})^T] A^T = A V A^T = D \quad D \text{ is a diagonal matrix}$$

- An orthonormal matrix  $A$  which diagonalizes the real symmetric covariance matrix  $V$  always exists.
- $A$  is the matrix of eigenvectors of  $V$ ,  $D$  is the matrix of corresponding eigenvalues.

$$\underline{x} = A^T \underline{y} + \underline{m} \quad \text{reconstruction of } \underline{x} \text{ from } \underline{y}$$

If  $\underline{x}$  is viewed as a point in  $n$ -dimensional Euclidean space, then  $A$  defines a rotated coordinate system.

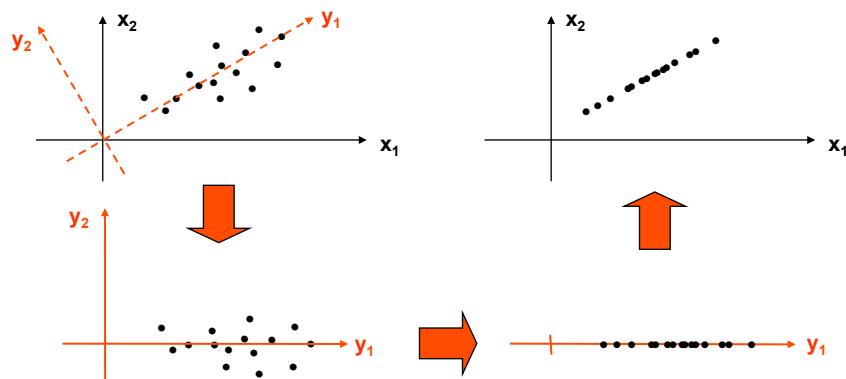
6

## Illustration of Minimum-loss Dimension Reduction

Using the Karhunen-Loëve transform, data compression is achieved by

- changing (rotating) the coordinate system
- omitting the least informative dimension(s) in the new coordinate system

Example:



7

## Compression and Reconstruction with the Karhunen-Loëve Transform

Assume that the eigenvalues  $\lambda_n$  and the corresponding eigenvectors in  $A$  are sorted in decreasing order  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$

$$D = \begin{bmatrix} \lambda_1 & 0 & 0 & \dots \\ 0 & \lambda_2 & 0 & \\ 0 & 0 & \lambda_3 & \\ \dots & & & \end{bmatrix}$$

Eigenvectors  $\underline{a}$  and eigenvalues  $\lambda$  are defined by  $V \underline{a} = \lambda \underline{a}$  and can be determined by solving  $\det[V - \lambda I] = 0$ .

There exist special procedures for determining eigenvalues of real symmetric matrices  $V$ .

Then  $\underline{x}$  can be transformed into a  $K$ -dimensional vector  $\underline{y}_K$ ,  $K < N$ , with a transformation matrix  $A_K$  containing only the first  $K$  eigenvectors of  $A$  corresponding to the largest  $K$  eigenvalues.

$$\underline{y}_K = A_K (\underline{x} - \underline{m})$$

The approximate reconstruction  $\underline{x}'$  minimizing the MSE is

$$\underline{x}' = A_K^T \underline{y}_K + \underline{m}$$

Hence  $\underline{y}_K$  can be used for data compression!

8

## Example for Karhunen-Loève Compression

$$N = 3 \quad V = \begin{bmatrix} 2 & -0,866 & -0,5 \\ -0,866 & 2 & 0 \\ -0,5 & 0 & 2 \end{bmatrix} \quad m = 0$$

$$\det(V - \lambda I) = 0 \quad \rightarrow \quad \lambda_1 = 3 \quad \lambda_2 = 2 \quad \lambda_3 = 1$$

$$A^T = \begin{bmatrix} 0,707 & 0 & 0,707 \\ -0,612 & 0,5 & 0,612 \\ -0,354 & -0,866 & 0,354 \end{bmatrix} \quad D = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Compression into K=2 dimensions:

$$\underline{y}_2 = A_2 \underline{x} = \begin{bmatrix} 0,707 & -0,612 & -0,354 \\ 0 & 0,5 & -0,866 \end{bmatrix} \underline{x}$$

Reconstruction from compressed values:

$$\underline{x}' = A_2^T \underline{y} = \begin{bmatrix} 0,707 & 0 \\ -0,612 & 0,5 \\ -0,354 & 0,354 \end{bmatrix} \underline{y}$$

Note the discrepancies between the original and the approximated values:

$$\begin{aligned} x_1' &= 0,5 x_1 - 0,43 x_2 - 0,25 x_3 \\ x_2' &= -0,085 x_1 - 0,625 x_2 + 0,39 x_3 \\ x_3' &= 0,273 x_1 + 0,39 x_2 + 0,25 x_3 \end{aligned}$$

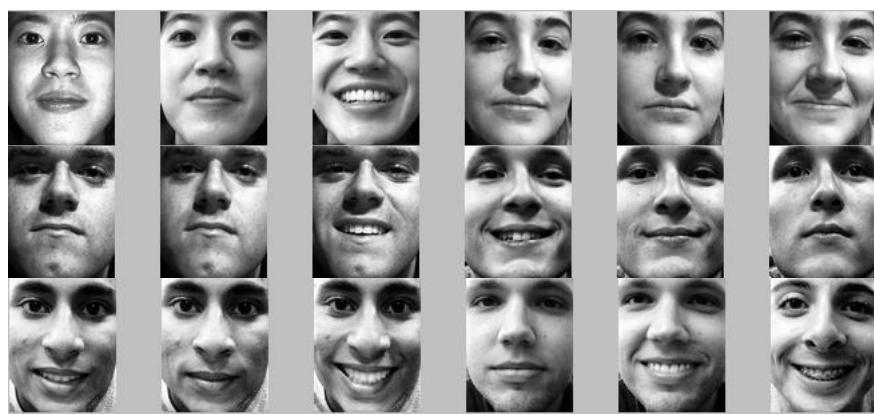
9

## Eigenfaces (1)

Turk & Pentland: Face Recognition Using Eigenfaces (1991)

Eigenfaces = eigenvectors of covariance matrix of normalized face images

Example images of eigenface project at Rice University



10

## Eigenfaces (2)

First 18 eigenfaces determined from covariance matrix of 86 face images



11

## Eigenfaces (3)

Original images and reconstructions from 50 eigenfaces



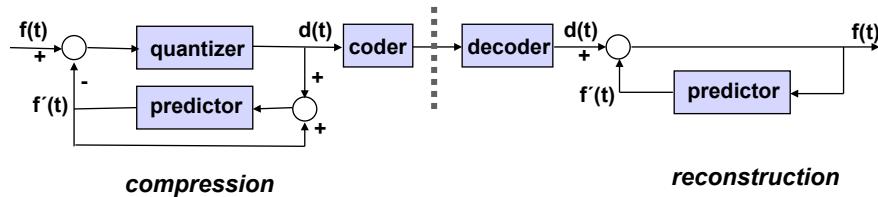
12

## Predictive Compression

Principle:

- estimate  $g_{mn}'$  from greyvalues in the neighbourhood of  $(mn)$
- encode difference  $d_{mn} = g_{mn} - g_{mn}'$
- transmit difference data (+ initially predictor)

For a 1D signal this is known as Differential Pulse Code Modulation (DPCM):



Linear predictor for a neighbourhood of K pixels:

$$g_{mn}' = a_1 g_1 + a_2 g_2 + \dots + a_K g_K$$

Computation of  $a_1 \dots a_K$  by minimizing the expected reconstruction error

13

## Example of Linear Predictor

For images, a linear predictor based on 3 pixels (3rd order) is often sufficient:

$$g_{mn}' = a_1 g_{m,n-1} + a_2 g_{m-1,n-1} + a_3 g_{m-1,n}$$

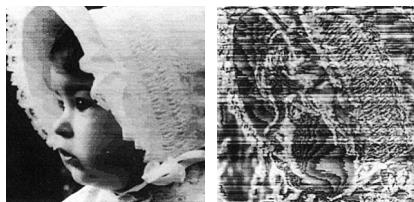
If  $g_{mn}$  is a zero mean stationary random process with autocorrelation  $C$ , then minimizing the expected error gives

$$\begin{aligned} a_1 C_{00} + a_2 C_{01} + a_3 C_{11} &= C_{10} \\ a_1 C_{01} + a_2 C_{00} + a_3 C_{10} &= C_{11} \\ a_1 C_{11} + a_2 C_{10} + a_3 C_{00} &= C_{01} \end{aligned}$$

n	01	11
00	10	

↑                      → m

This can be solved for  $a_1, a_2, a_3$  using Cramer's Rule.



Example:

Predictive compression with 2nd order predictor and Huffman coding, ratio 6.2

Left: Reconstructed image

Right: Difference image (right) with maximal difference of 140 greylevels

14

## Discrete Cosine Transform (DCT)

Discrete Cosine Transform is commonly used for image compression,  
e.g. in JPEG (Joint Photographic Expert Group) Baseline System standard.

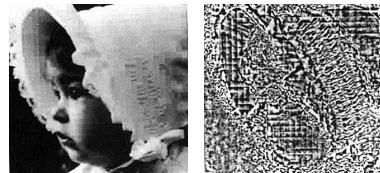
**Definition of DCT:**  $G_{00} = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} g_{mn}$

$$G_{uv} = \frac{1}{2N^3} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} g_{mn} \cos[(2m+1)u\pi] \cos[(2n+1)v\pi]$$

**Inverse DCT:**  $g_{mn} = \frac{1}{N} G_{00} + \frac{1}{2N^3} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} G_{uv} \cos[(2m+1)u\pi] \cos[(2n+1)v\pi]$

In effect, the DCT computes a Fourier Transform of a function made symmetric at N by a mirror copy.

=> 1. Result does not contain sinus terms 2. No wrap-around errors



Example:

DCT compression with ratio 1 : 5.6

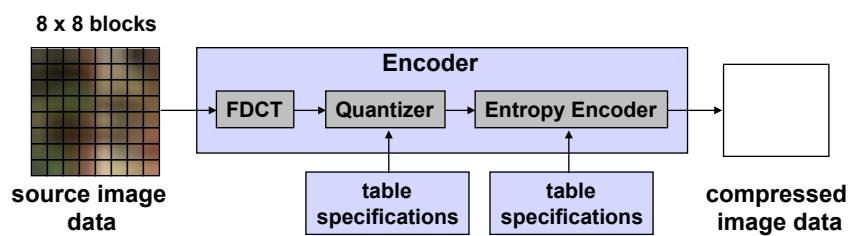
Left: Reconstructed image

Right: Difference image (right) with maximal difference of 125 greylevels

15

## Principle of Baseline JPEG

(Source: Gibson et al., Digital Compression for Multimedia, Morgan Kaufmann 98)



- transform RGB into YUV coding, subsample color information
- partition image into 8 x 8 blocks, left-to-right, top-to-bottom
- compute Discrete Cosine Transform (DCT) of each block
- quantize coefficients according to psychovisual quantization tables
- order DCT coefficients in zigzag order
- perform runlength coding of bitstream of all coefficients of a block
- perform Huffman coding for symbols formed by bit patterns of a block

16

## YUV Color Model for JPEG

Human eyes are more sensitive to luminance (brightness) than to chrominance (color). YUV color coding allows to code chrominance with fewer bits than luminance.

CCIR-601 scheme:

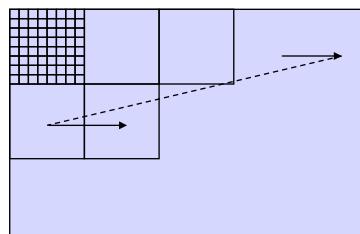
$$\begin{aligned} Y &= 0.299 R + 0.587 G + 0.144 B && \text{"luminance"} \\ Cb &= 0.1687 R - 0.3313 G + 0.5 B && \text{"blueness"} \rightarrow \square \square \\ Cr &= 0.5 R - 0.4187 G - 0.0813 B && \text{"redness"} \rightarrow \square \square \end{aligned}$$

In JPEG:

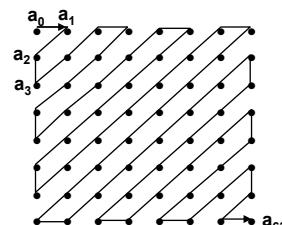
1 Cb, 1 Cr and 4 Y values for each  $2 \times 2$  image subfield  
(6 instead of 12 values)

17

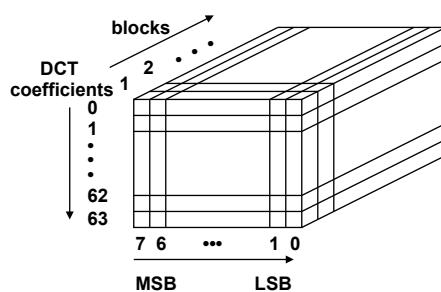
## Illustrations for Baseline JPEG



partitioning the image into blocks



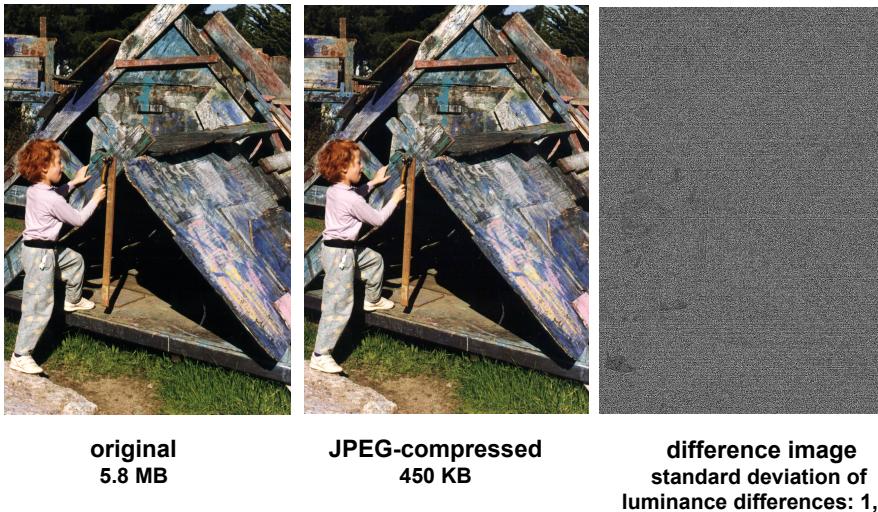
DCT coefficient ordering for efficient runlength coding



transmission sequence for blocks of image

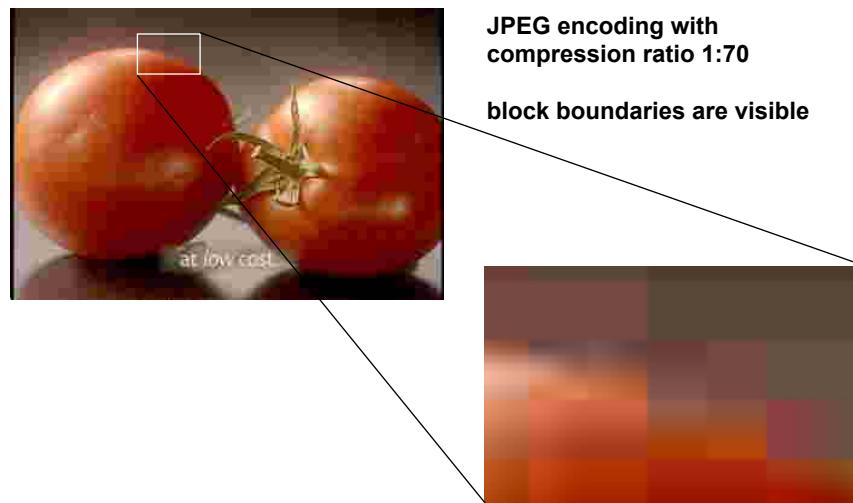
18

## JPEG-compressed Image



19

## Problems with Block Structure of JPEG



20

10

## Progressive Encoding

Progressive encoding allows to first transmit a coarse version of the image which is then progressively refined (convenient for browsing applications).

### Spectral selection

1. transmission: DCT coefficients  $a_0 \dots a_{k_1}$
2. transmission: DCT coefficients  $a_{k_1} \dots a_{k_2}$
- :

low frequency  
coefficients first

### Successive approximation

1. transmission: bits  $7 \dots n_1$
2. transmission: bits  $n_1+1 \dots n_2$
- :

most significant  
bits first

21

## MPEG Compression

### Original goal:

Compress a 120 Mbps video stream to be handled by a CD with 1 Mbps.

### Basic procedure:

- temporal prediction to exploit redundancy between image frames
- frequency domain decomposition using the DCT
- selective reduction of precision by quantization
- variable length coding to exploit statistical redundancy
- additional special techniques to maximize efficiency

### Motion compensation:

16 x 16 blocks luminance with 8 x 8 blocks chromaticity of the current image frame are transmitted in terms of

- an offset to the best-fitting block in a reference frame (motion vector)
- the compressed differences between the current and the reference block

22

11

## MPEG-7 Standard

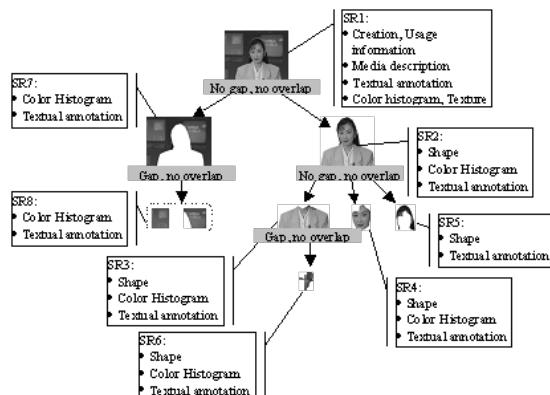
**MPEG-7: “Multimedia Content Description Interface”**

- introduced as standard in 2002
- supports multimedia content description (audio and visual)
- not aimed at a particular application

Description of visual contents in terms of:

- descriptors (e.g. color, texture, shape, motion, localization, face features)
- segments
- structural information
- Description Definition Language (DDL)

↓  
segmentation methodology required!

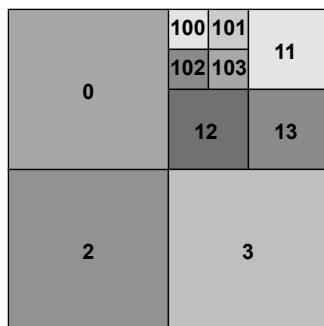


23

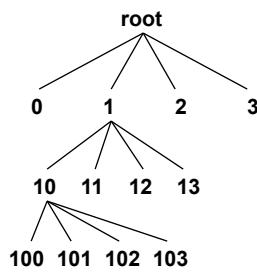
## Quadtree Image Representation

Properties of quadtree:

- every node represents a squared image area, e.g. by its mean greyvalue
- every node has 4 children except leaf nodes
- children of a node represent the 4 subsquares of the parent node
- nodes can be refined if necessary



quadtree structure:



24

## Quadtrees Image Compression

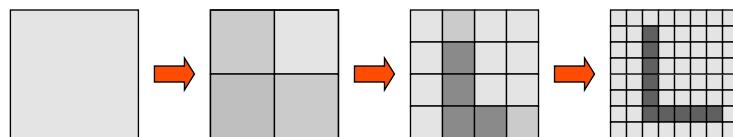
A complete quadtree represents an image of  $N = 2^K \times 2^K$  pixels with  
 $1 + 4 + 16 + \dots + 2^{2K}$  nodes  $\approx 1.33 N$  nodes.

An image may be compressed by

- storing at every child node the greyvalue difference between child and parent node
- omitting subtrees with equal greyvalues

Quadtree image compression supports progressive image transmission:

- images are transmitted by increasing quadtree levels, i.e. images are progressively refined
- intermediate image representations provide useful information, e.g. for image retrieval



25