Adaptive transform of the color space in image compression

Kirill Yourkov

March 31th 2005

Image compression applications.

- Digital television
- Digital photography
- Systems for digital transmission and storage of video signals via standard telephone channels
- Many others

Examples

- Let consider data transmission via telephone channel using standard modem with rate 3360 b/s. If we need to transmit an image of size 240x352 points and for each point representation we need 3 bytes than it will take about 1 minute for transmission
- One minute of full-screen video with 480x720 resolution and 30 frames per second speed needs 20.736 Mb.

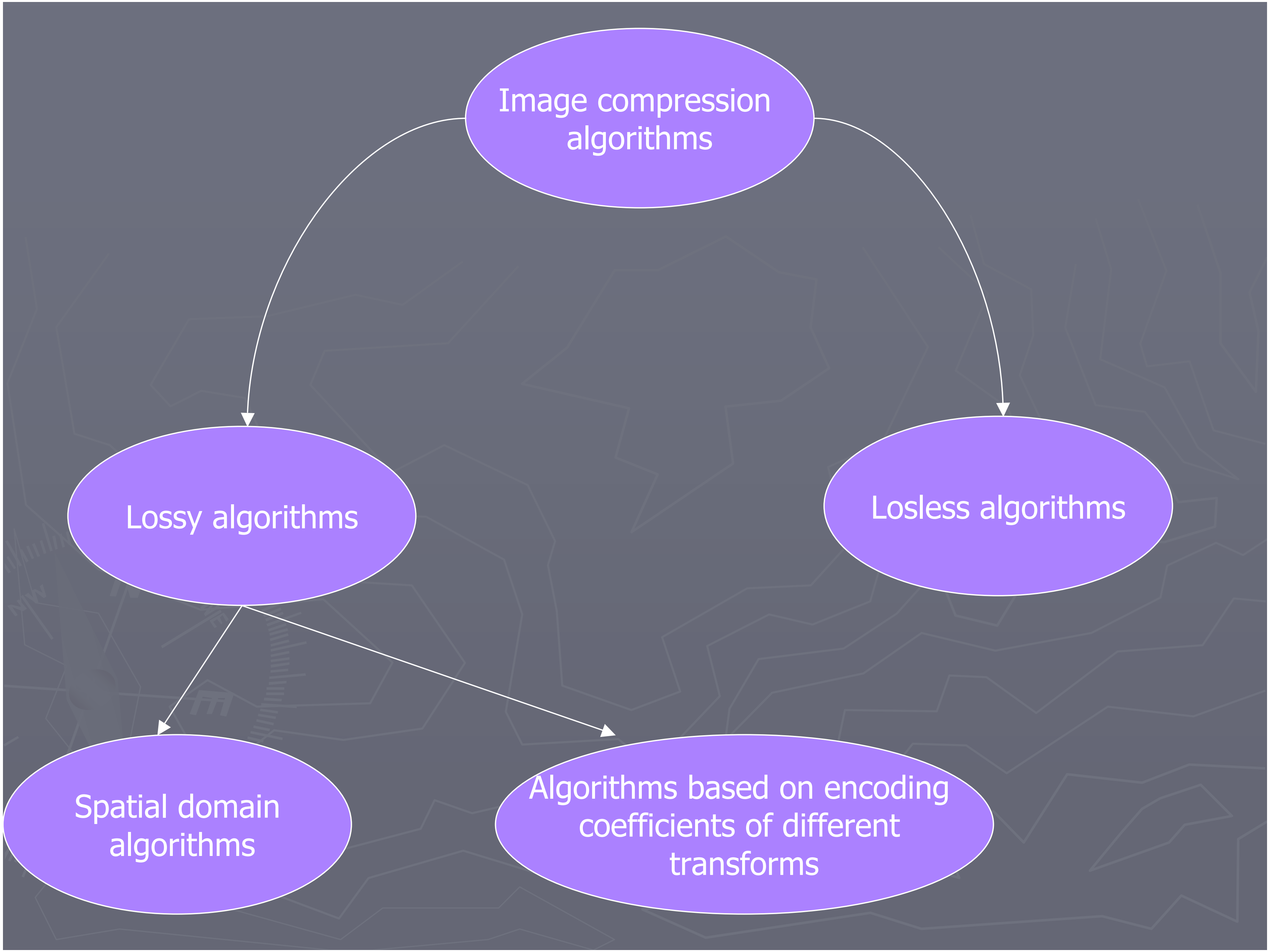


Image compression algorithms
Lossy algorithms
Losless algorithms
Spatial domain algorithms
Algorithms based on encoding coefficients of different transforms

The criteria for algorithm performance estimation

- Encoding efficiency (bit rate)

  Usually measured in bits per sample or in bits per second

- Quality of restored image

  Usually it's signal to noise ratio (SNR) for decoder output

$$SNR = 10\log_{10}\left(\frac{E_{inp}}{E_N}\right), \quad (Db)$$

E_{inp}- energy of encoder input signal E_n - energy of noise signal

For video and image compression algorithms quality measure is:

$$PSNR = 10\log_{10}\left(\frac{255^2}{E_N}\right), \quad (Db)$$

- Encoding complexity
- Encoding delay

Digital image representation

- *BMP* (bit map)

  It is one of the main formats in color image representation. The standard representation of pixels in this format is RGB, it uses 24 bits for one pixel. Every color component (Red, Green, Blue) is represented by one byte.

  If the image has size $n^{x}m$ then 3*n*m bits are needed for image representation.

Main stages of image compression algorithms based on block transform

1. Color space transform
2. 2-Dimensional orthogonal linear transform on blocks of color components
3. Quantization
4. Scanning quantized 2-d data in some order
   - Run-length coding
5. Entropy encoding of quantized values

Color space transforms

- Motivation:
  1. Correlation between color components
  2. Energy overlapping
  3. Human visual color perception
- Sample types of color spaces:
  1. RGB
  2. YUV
  3. YCbCr
  4. YIQ

  Etc.

RGB - YUV transform

RGB-YUV – the main transform which used in image compression. This is linear transform with fixed coefficients.

Forward transform:

$$Y = 0.299R + 0.587G + 0.144B$$

$$U = (B - Y)0.5643 + 128$$

$$V = (R - Y)0.7132 + 128$$

Inverse transform:

$$G = Y - 0.714(V - 128) - 0.334(U - 128)$$

$$R = Y + 1.402(V - 128)$$

$$B = Y + 1.772(U - 128)$$

2-Dimensional transforms

- Desirable properties of transform:
  1. Most energy localization in several numbers of output coefficients.
  2. Decorrelation of output coefficients.
  3. Orthonormal property.
  4. Low computational complexity. In particular separation property.

Let $T_c = \{t_c(u,i)\}, u,i = 0,1 \ldots N-1$
$T_r = \{t_r(v,j)\}, u,i = 0,1 \ldots N-1$ be transform matrices.

Than 2-dimensional transform associated with this matrices is:

$$Y = T_c X T_r^{'}$$

DCT

Discrete Cosine Transform – the main transform used for image compression:

•For images with large correlation between pixels this transform good approximation of Kharunen-Lovev transform.
•Because of independence on the image, separate property, orthonormal property, calculation complexity is rather low.

$$S(v,u) = \frac{C(v)}{2}\frac{C(u)}{2}\sum_{y=0}^{7}\sum_{x=0}^{7} s(y,x)\cos[(2x+1)u\pi/16]\cos[(2y+1)v\pi/16]$$

$$s(y,x) = \sum_{v=0}^{7}\frac{C(v)}{2}\sum_{u=0}^{7}\frac{C(u)}{2} S(v,u)\cos[(2x+1)u\pi/16]\cos[(2y+1)v\pi/16]$$

$$C(u) = 1/\sqrt{2}, \quad u = 0 \qquad C(v) = 1/\sqrt{2}, \quad v = 0$$

$$C(u) = 1, \quad u > 0 \qquad C(v) = 1, \quad v > 0$$

Quantization

- This part of encoding introduce an error
- The more error we introduce the less bits we need for image description and vice versa
- Usually scalar quantization applied because of its simplicity

$$z_{kl} = \lfloor (y_{kl} + sign(y_{kl})q/2)/q \rfloor, k,l = 0,1,\ldots 7$$

z_{kl} - quantized value,

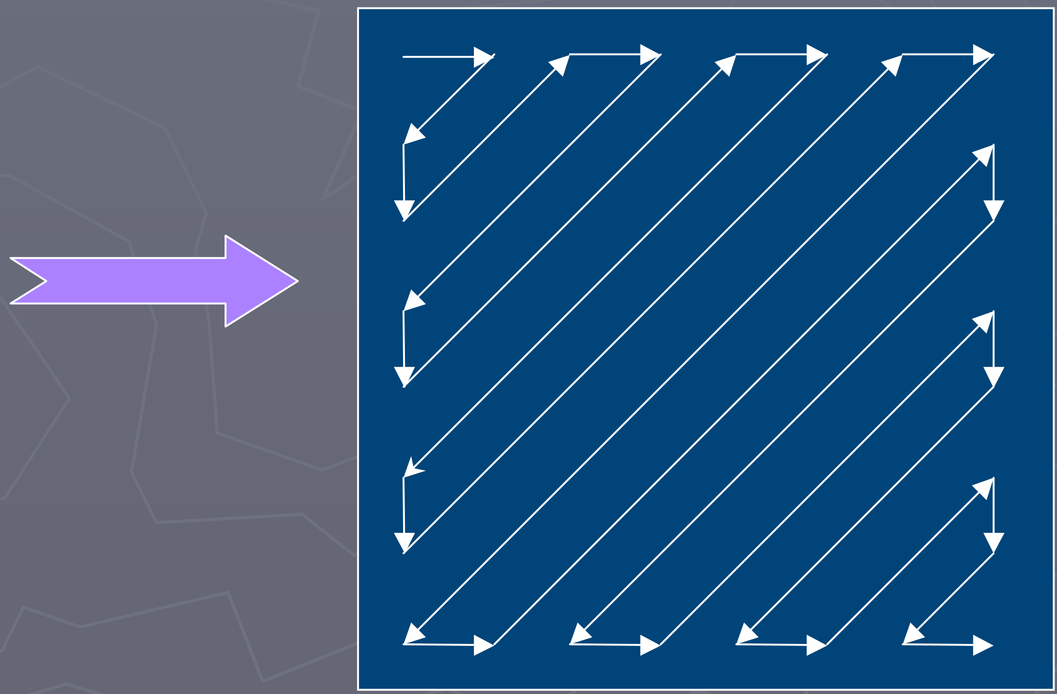
y_{kl} - input value.

q - quantization step.

Scanning order

- For quantized DCT coefficients such scanning order is used:

Zigzag scanning order



12	12	12	12	12	12	12	12
12	12	12	12	12	11	11	11
12	12	13	13	13	13	13	13
29	27	30	35	44	52	60	64
43	38	36	40	51	62	72	78
50	44	39	41	47	56	67	75
48	42	40	42	47	54	66	72
41	40	42	44	46	50	61	69

DCT

284	–54	24	1	3	1	0	1
132	31	–19	1	–2	–1	0	0
–45	24	–6	–5	0	0	0	0
30	–5	9	3	0	0	0	0
28	–17	0	0	0	0	0	0
3	–3	0	0	0	0	0	0
–15	8	–1	–1	0	0	0	0
–11	4	0	1	0	0	0	0

Quantization q=14

36	–4	2	0	0	0	0	0
–9	2	–1	0	0	0	0	0
–3	2	0	0	0	0	0	0
2	0	1	0	0	0	0	0
2	–1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
–1	1	0	0	0	0	0	0
–1	4	0	0	0	0	0	0

DC coefficient

Zigzag scan

36 -4 -9 -3 2 2 0 -1 2 2 2 0 0 0 0 0 0 0 1 -1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 4 0 0 .....

Run-level encoding

36 (-4 0) (-9 0) (-3 0) (2 0) (2 1) (-1 0) (2 0) (2 0) (2 7) (1 0) (-1 1) (-1 13) (-1 0) (4 ...)

Entropy encoding

- Representing values with different probability appearance with different number of bits is the main idea of entropy coding
- Techniques usually used for entropy coding:
  1. Arithmetic coding
  2. Huffman coding
  3. Ziv-Lempel class algorithms, ets.

Image size: 320x240, 28.8KB

Ratios (original : compressed):

1. 1:3
2. 1:18
3. 1:30
4. 1:50
5. 1:104

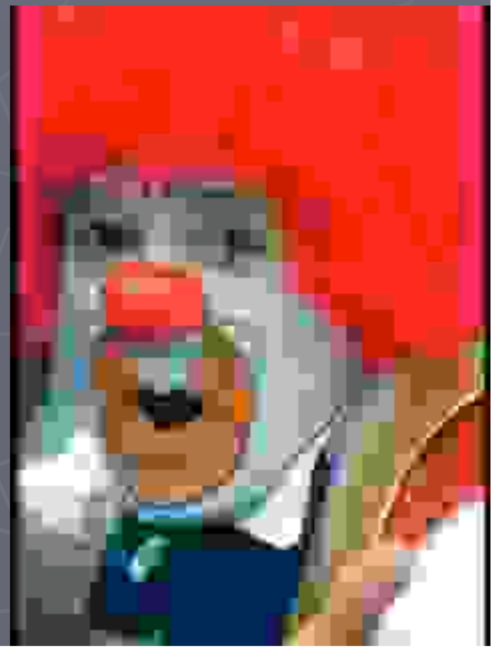
2

1

3

4

5



One approach to color space transform in terms of image compression

- Setting mathematical problem:

$$\arg\min_{L} \; g(L(IMG))$$

Where g – criteria-function, L – linear transform.

Let consider a block of *n* points:

n points

R	$r_1 \; r_2 \; r_3 \; \ldots \; r_n$
G	$g_1 \; g_2 \; g_3 \; \ldots \; g_n$
B	$b_1 \; b_2 \; b_3 \; \ldots \; b_n$

$:= X$

For this block:

$$K_X = XX^T$$ – correlation matrix.

We are to find matrix A, such as $D := AXX^T A^T$ will diagonal.

Solution of the problem

Matrix K_X is symmetric matrix, so it can be represented as:

$$K_X = U \times \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \times U^T,$$

where U – orthogonal matrix, which includes eigen vectors of K_X .

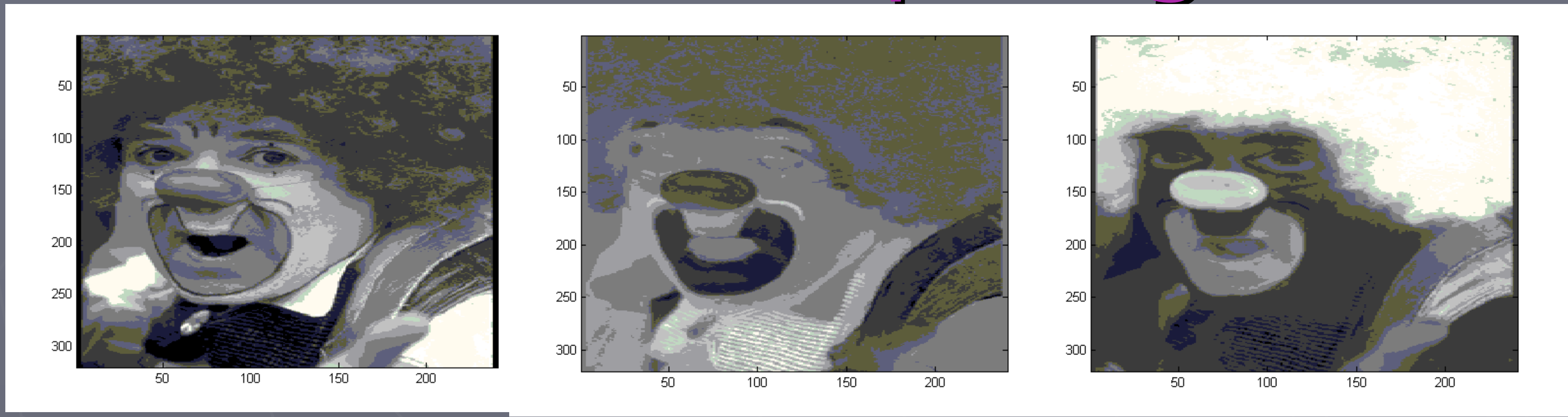
$\lambda_1, \lambda_2, \lambda_3$ – are eigen values of K_X .

If we take as transform matrix $A := U^T = U^{-1}$ then:

$$D = AXX^T A^T = U^T K_X U = U^T U \Lambda U^T U = \Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

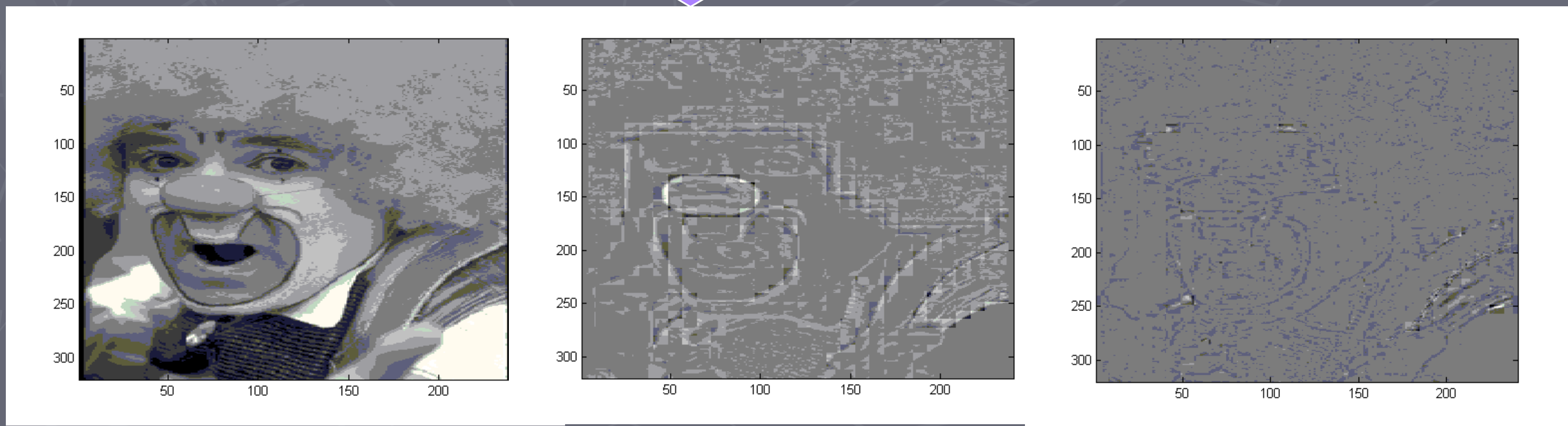
And we have the solution for problem putted by.

Visual comparing

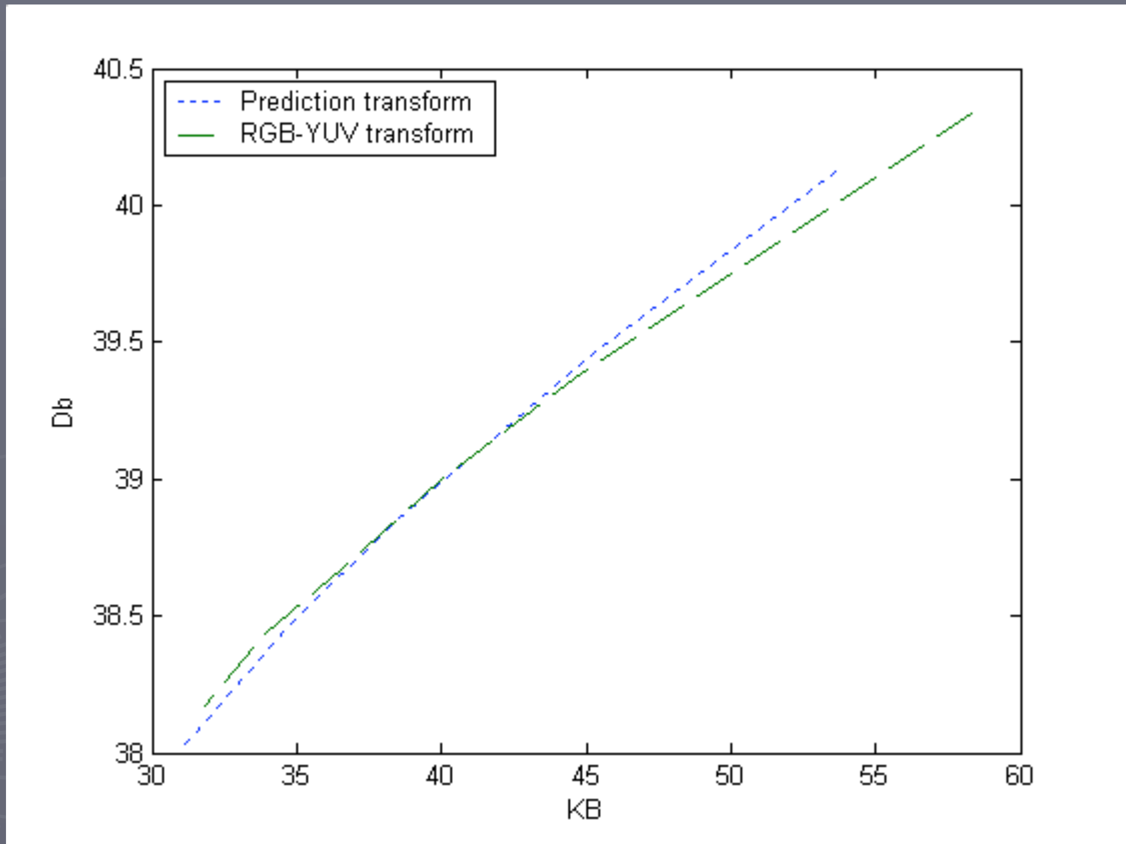


Y,U,V components

Optimal transform



Bits results



Thank you for attention!
Any questions?