Algorithm for encoding (simple version)

(Assuming gray scale images, each pixel represents a value from 0 to 255 the intensity of the color)

Domain (i,j)

$i, j \in [1, 16]$

64 x 64

Range (k,l)

$k, l \in [1, 32]$

128 x 128

Now applying domain transformation (T)

$$T(D_{i,j}) = \alpha D_{i,j} + t_0$$

$T(D_{i,j}) \geq 0$

$\alpha \in [0,1]$ and $t_0 \in [-255, 255]$

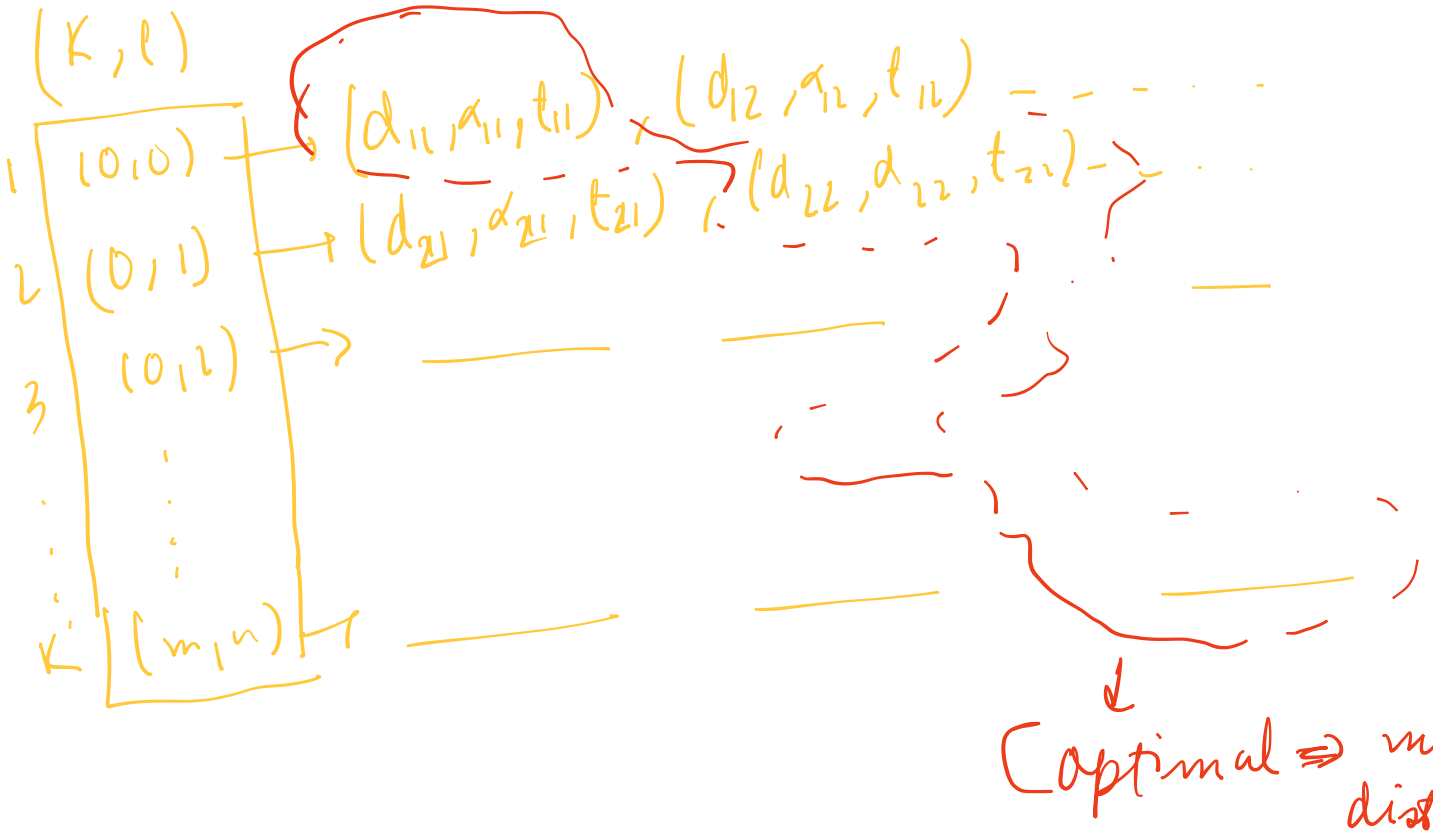
Now, we apply a Brute force algorithm to map domain (i,j) to range (k,l)

Take (k,l) from range:

Take α from $[0,1]$:

Take t from $[-255, 255]$:

Apply the affine transformation to all (i,j) in domain and take the (i,j) for which difference in pixel value is min^n for (k,l) and store in a list.



Now, we need to choose transformation such that distortion is minimized :-

$$\text{distortion } (d) = \sum \{T(D_{i,j}) - R_{k,l}\}^2$$

Minimize this

Now, that we have the optimal transformations we can decode it from here. (These are stored in Fractal Code Book)

Algorithm for decoding (simple version) :-

Take some initial image Ω_{init}

$$\Omega_1 = \eta(\Omega_{init})$$

$$\Omega_2 = \eta(\Omega_1)$$

$$\Omega_3 = \eta(\Omega_2)$$

$$\vdots$$

$$\Omega_n = \eta(\Omega_{n-1}) \rightarrow \text{converge to original}$$

$$\eta = \psi(\Omega)\, T(\Omega)$$

$T \rightarrow$ affine transformation

$\psi \rightarrow$ down sampling + low pass filtering