Codificação e Compressão

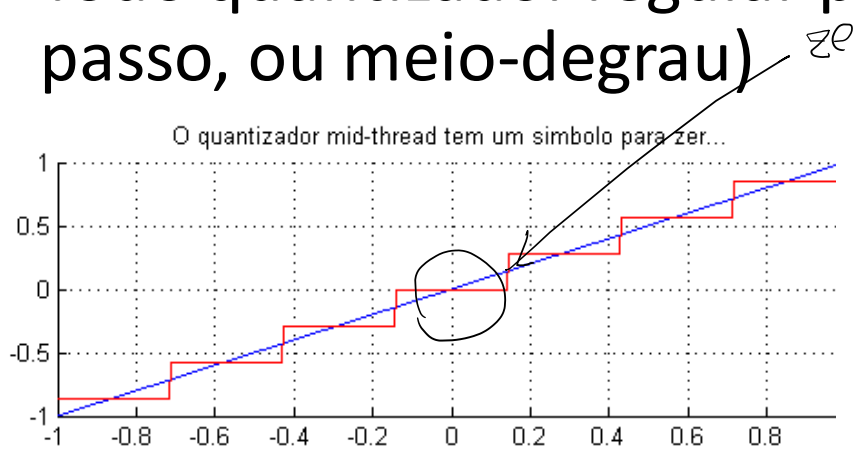
Quantização de Imagem, vetorial e transformadas

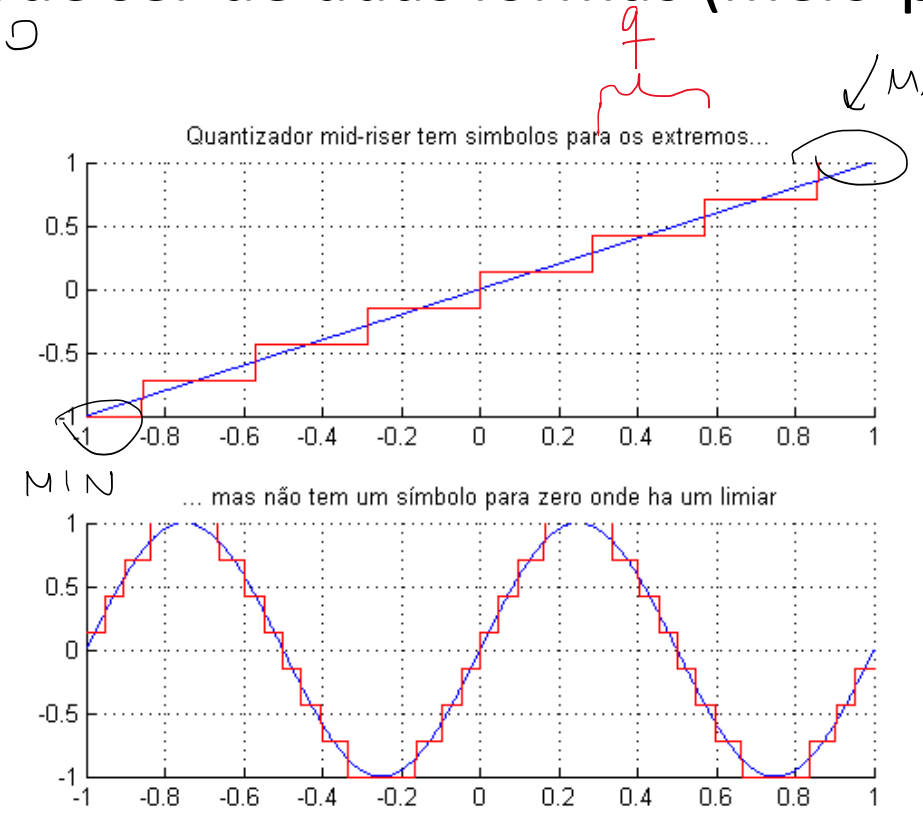
Na aula passada

- Quantização Uniforme
- Quantização Não Uniforme
  - Uniformização de PDF
  - Companding independente de PDF
  - Algoritmo de Lloyds

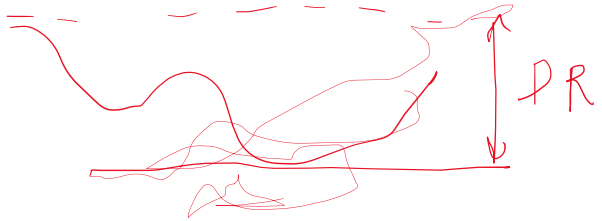
Quantizadores uniformes

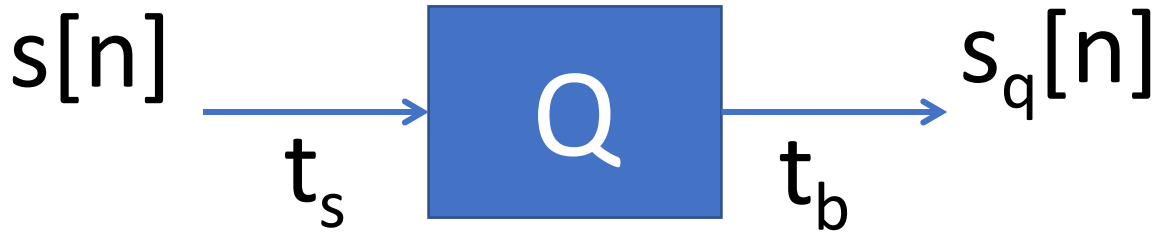
- Todo quantizador regular pode ser de duas formas (meio-piso e meio-passo, ou meio-degrau)





Quantizadores uniformes





$$t_b = r \cdot t_s$$

Para um quantizador uniforme o passo de quantização q é:

$$q = \frac{DR}{M-1} = \frac{DR}{2^r - 1} \rightarrow 2^r = \frac{DR}{q} + 1 \sim \frac{DR}{q}$$

Onde M é o número de símbolos no seu codebook.

Dependendo se o quantizador for meio-piso ou meio-degrau os valores de seu codebook serão diferenetes.

Quantizador uniforme

- Os valores de codebook de um quantizador uniforme meio piso (ou seja há um código para zero – “mid-thread”)são dados por:

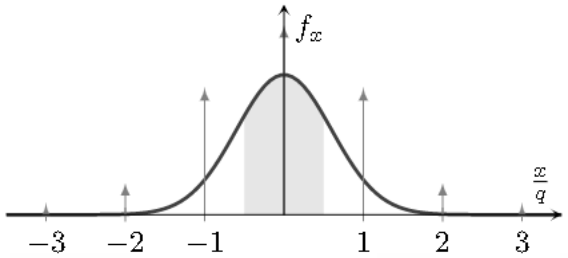
$$x_q = q \cdot \left(\left\lfloor \frac{x}{q} \right\rfloor + \frac{1}{2} \right)$$

- Já no caso do quantizador uniforme de meio passo (mid-step) são dados pela expressão abaixo:

$$x_q = q \cdot \lfloor \frac{x}{q} + \frac{1}{2} \rfloor$$

Ruido de quantização

- A distribuição de um sinal quantizado uniformemente pode ser derivada de sua distribuição de entrada:



- Pode-se ver que:

$$f_{x_q} = \sum_{i=-M/2, i\neq 0}^{i=M/2} \left(\int_{q\cdot i - q/2}^{q\cdot i + q/2} f_x(x)dx \right) \cdot \delta(x - i\cdot q)$$

- De onde:

$$f_{x_q} = q \cdot rect\left(\frac{x}{q}\right) \odot f_x \cdot \sum_{i=-M/2, i\neq 0}^{i=M/2} \delta(x - i\cdot q)$$

QT I e II

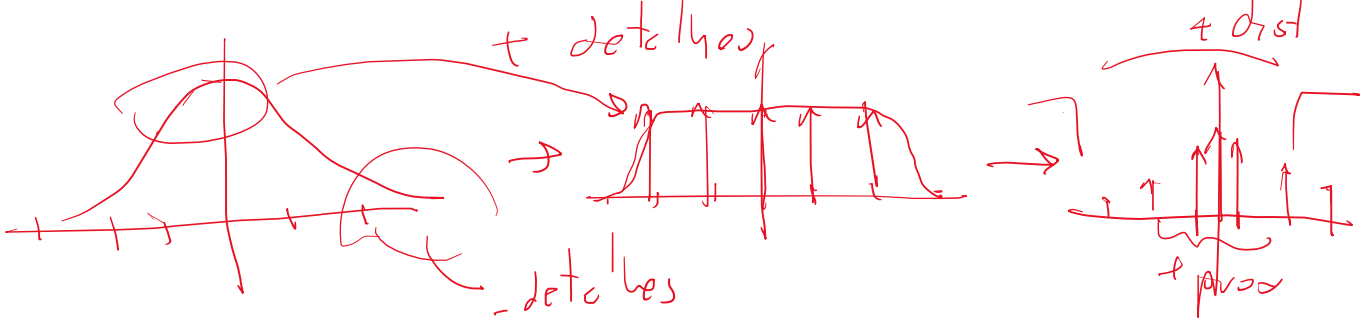
Essa expressão é similar a da amostragem e nos gera uma espécie de equivalente ao teorema de Nyquist na forma dos teoremas de quantização. Considerando W a largura da função característica de x (ou seja CF=0 para valores maiores que W)

- QT I: É possível reconstruir a PDF de x se $\frac{1}{q} \geq 2 \cdot W$

- QT II: : É possível reconstruir os momentos de x se $\frac{1}{q} \geq W$

Observer que, para um sinal gaussiano $DR \geq \frac{6}{W}$ sempre ou haverá saturação.

Companding



- No caso do companding a ideia é aplicar uma função não linear ao sinal antes de passar por um quantizador uniforme:

- A questão é qual a função não linear que se deve utilizar...
- Para isso há duas abordagens:
  - Caso se conheça a PDF de x e ela seja bastante estável ao longo do tempo se pode definir uma função sob medida
  - Caso não se conheça a PDF se pode aproximar um companding independente de PDF

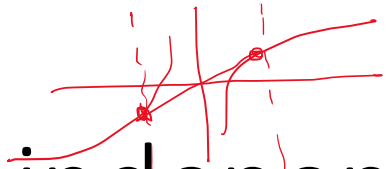
Companding independente de PDF

Segundo Bennets et al: Assumindo R≥ 6, pode-se dizer que:

$$\sigma_q^2 \approx \frac{q^2}{12} \int_{-x_{MAX}}^{+x_{MAX}} \frac{f_x(x)}{\left|\frac{\partial g(x)}{\partial x}\right|^2} \mathrm{dx}$$

Lembrando que para y = g(x) sabe-se que a $f_y = \left|\frac{d}{dy} g^{-1}(x)\right| f_x(y)$

Companding independente de PDF



$$\sigma_x^2 = \int_{-\infty}^{\infty} x^2 f_x \, dx$$

$$SNR_Q = \frac{\sigma_x^2}{\sigma_q^2}$$

$$\frac{1}{\left|\frac{\partial g(x)}{\partial x}\right|^2} = k \cdot x^2 \qquad k = \frac{1}{k'} \qquad k' = \sqrt{k}$$

$$\left|\frac{\partial g(x)}{\partial x}\right|^2 = \frac{k'}{x^2} \qquad \left|\frac{\partial g(x)}{\partial x}\right| = \frac{k}{x}$$

Aplico

$$SNR_Q = \frac{\int_{-\infty}^{\infty} x^2 f_x \, dx}{\frac{q^2}{12} \int_{-x_{MAX}}^{x_{MAX}} \left(\frac{f_x}{\left|\frac{\partial g(x)}{\partial x}\right|^2} \right) dx}$$

P/ eliminar f_x

$$\left|\frac{\partial g(x)}{\partial x}\right|^2 = \left|\frac{k}{x}\right|^2$$

$$\frac{k}{x} \qquad \frac{F_x}{\frac{k^2}{x^2}} = \frac{x^2 f_x}{x^2} \qquad \left|\frac{\partial g(x)}{\partial x}\right| = \frac{k}{x}$$

Companding independente de PDF

- Dai se deduzem expressões como a lei u e a lei A

$$F(x) = \operatorname{sgn}(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)}, \quad -1 \leq x \leq 1$$

Lei μ

Para 8 bits μ = 255

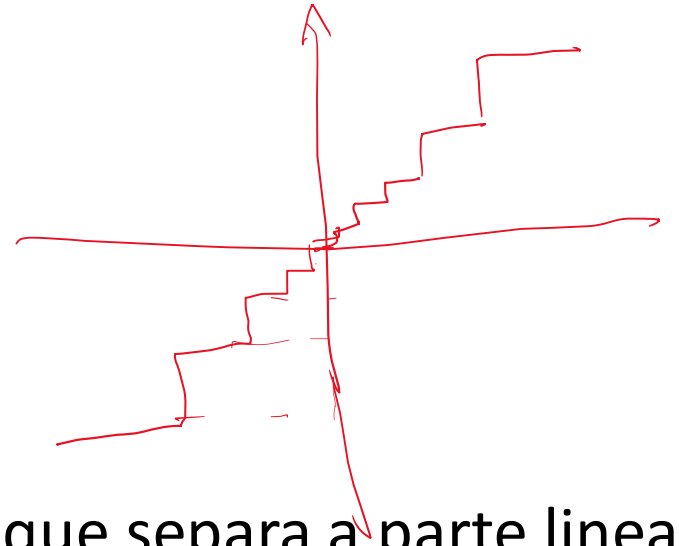
$$F(x) = \operatorname{sgn}(x)\begin{cases} \dfrac{A|x|}{1+\ln(A)}, & |x| < \dfrac{1}{A} \\ \dfrac{1+\ln(A|x|)}{1+\ln(A)}, & \dfrac{1}{A} \leq |x| \leq 1, \end{cases}$$

Lei A

Para 8 bits A = 87,56

Algoritmo de Lloyds

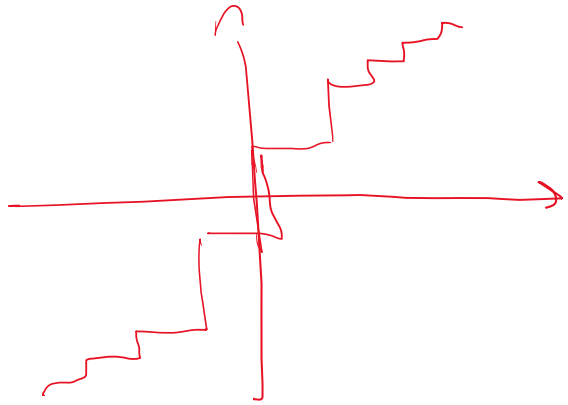
$$\sigma_q^2 = E\{|x - x_q|^2\} = 2\int_0^{\infty} e^2(x) f_x \, dx$$



- Se o quantizador não for linear existe um ponto v que separa a parte linear da saturação não linear

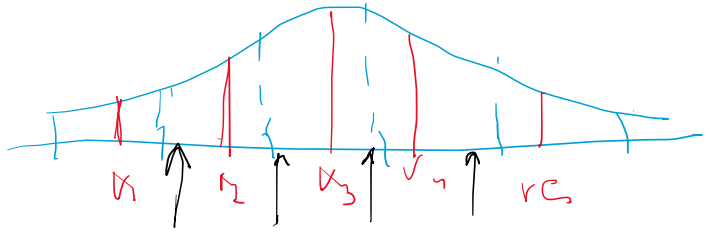
$$\sigma_D^2 = 2\int_0^{v} e^2(x) f_x \, dx + 2\int_v^{\infty} e^2(x) f_x \, dx$$

$$\sigma_D^2 = \sigma_q^2 + \sigma_s^2$$

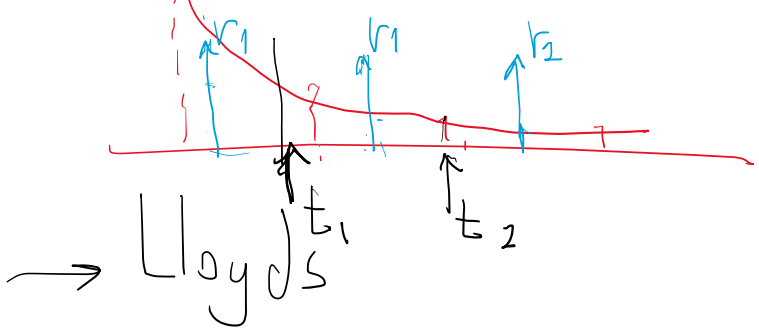


- Cada intervalo de quantização adiciona uma contribuição diferente ao erro que depende da PDF

Algoritmo interativo



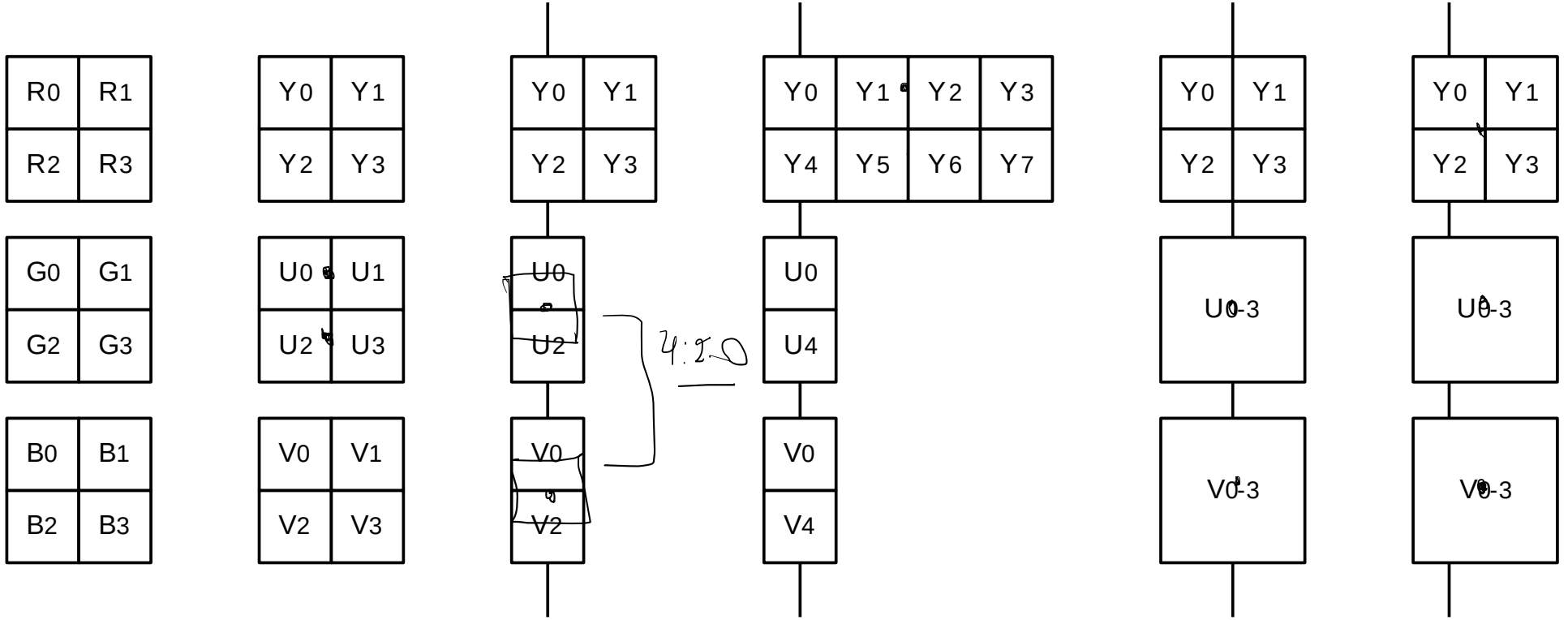
1. Entre R inicial, interações I, faixa dinâmica mínima t_1 e máxima t_N
2. Gere thresholds iniciais t_1^0,... t_N^0 [por exemplo, inicia uniforme]
3. Enquanto interações < I
   a) Usa t inicial e gera histograma do treinamento
   b) Calcula o MSE entre o sinal original e o quantizado dessa forma
   c) Para cada intervalo, calcula o centro de gravidade e divide pelo número de amostras no intervalo. Isso gera r_n
   d) Recalcula os $t_n^{opt} = \left(r_n^{opt} + r_{n+1}^{opt}\right)/2$
   e) Avalia a SNR



Quantização de Imagens

Imagens são divididas em blocos

Esses blocos podem passar por transformação de cor e subamostragem de pixeis primeiro

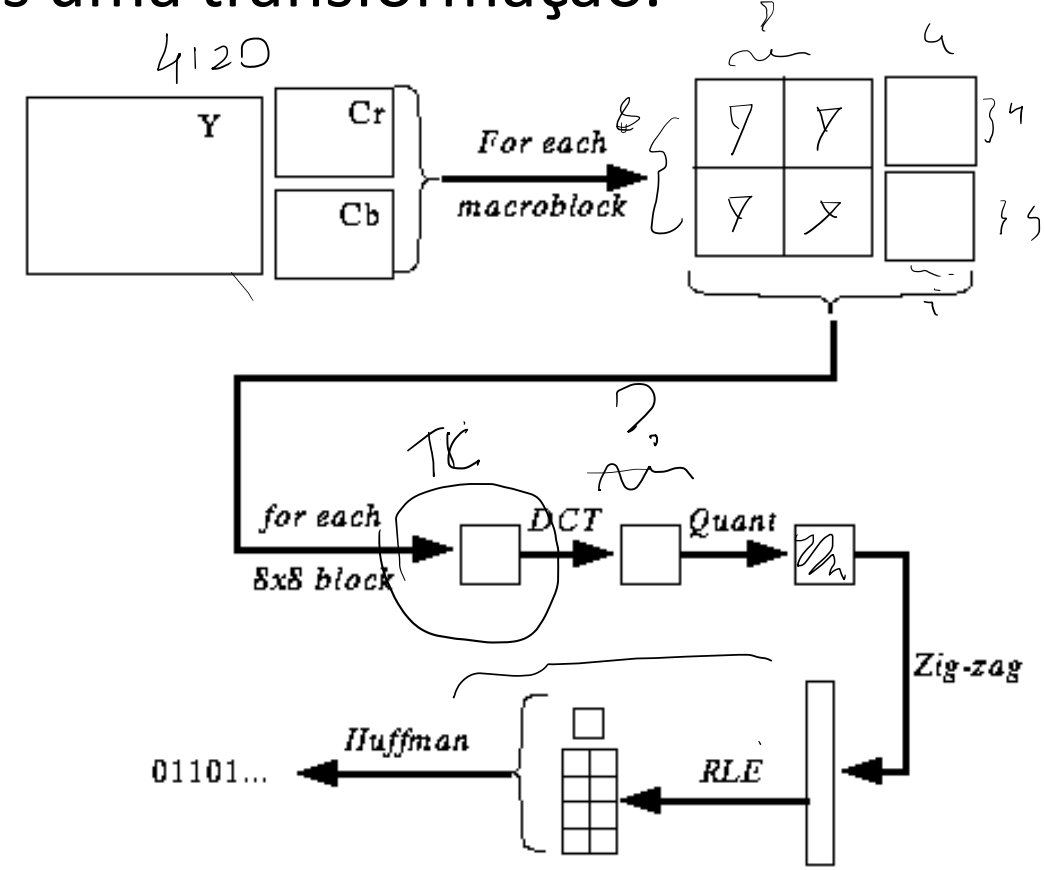


RGB | 4:4:4 | 4:2:2 | 4:1:1 | 4:2:0 JPEG | 4:2:0 MPEG-2

A quantização vem logo após

- Como veremos depois o que é quantizado podem ser os valores de cada bloco ou os coeficientes após uma transformação.

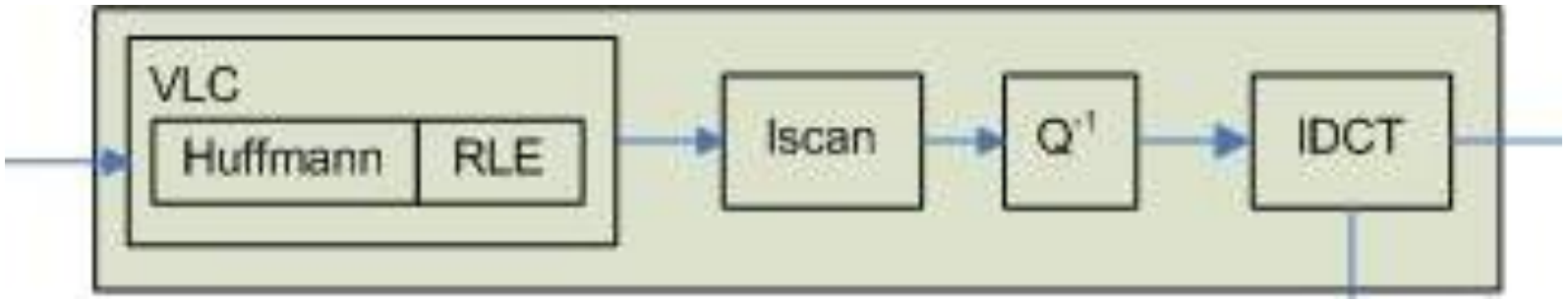
Ex: Esquema para MPEG/JPEG



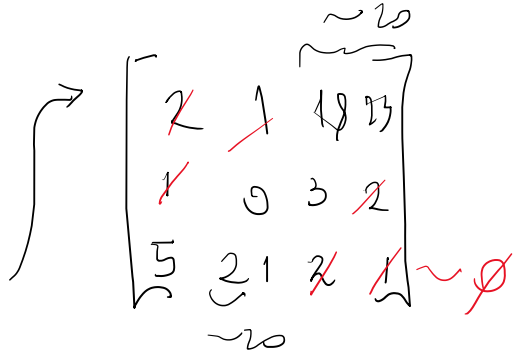
PNG: deflate + huffman (LZ77)

Na decodificação o processo é revertido

- Para casos com transformação antes (por exemplo):



- Note que vários elementos podem mudar
  - VLC diferente
  - Não usar transformada
- No entanto, a quantização tem de ser feita **por blocos**

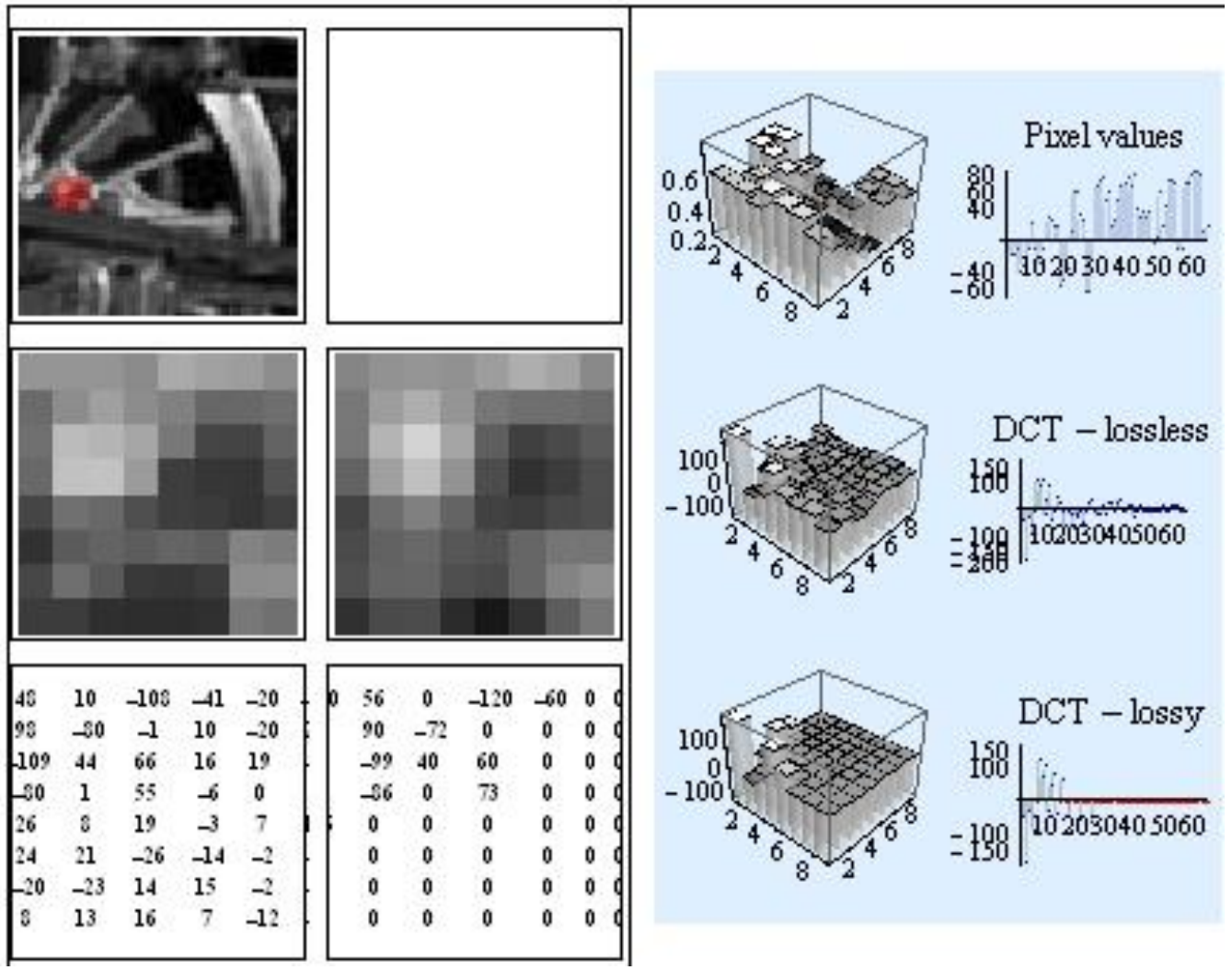


Quantizador

- A quantização é feita, nesse caso, dividindo-se o bloco por uma matriz que busca maximizar o número de coeficientes zerados

Q =

16	19	22	26	27	29
16	22	24	27	29	34
22	26	27	29	34	34
22	26	27	29	34	37
26	27	29	32	35	40
27	29	32	35	40	48
27	29	34	38	46	56
29	35	38	46	56	69



Em resumo

- O processo de quantização é realizado elemento a elemento na matriz que representa o bloco
- Para cada elemento do bloco existe um fator de divisão e a saída calculada é dada por:

$$c_{ij}^{q} = \left\lfloor \frac{c_{ij}}{Q_{ij}} \right\rfloor$$

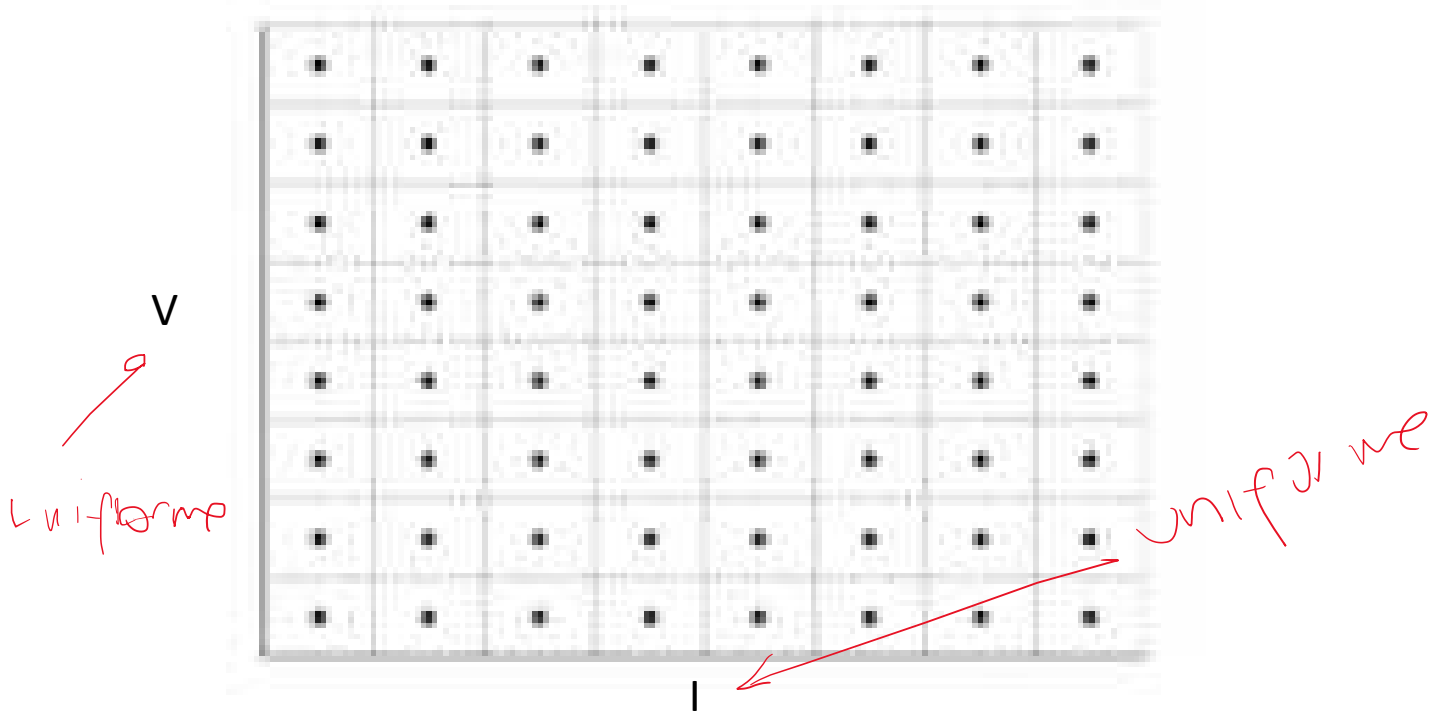
- De mesma forma na decodificação os valores são multiplicados pelos mesmos fatores.
- A quantização inversa é, de fato, um processo de reescalonamento.

Quantização Vetorial

Indo além da quantização de uma única variável

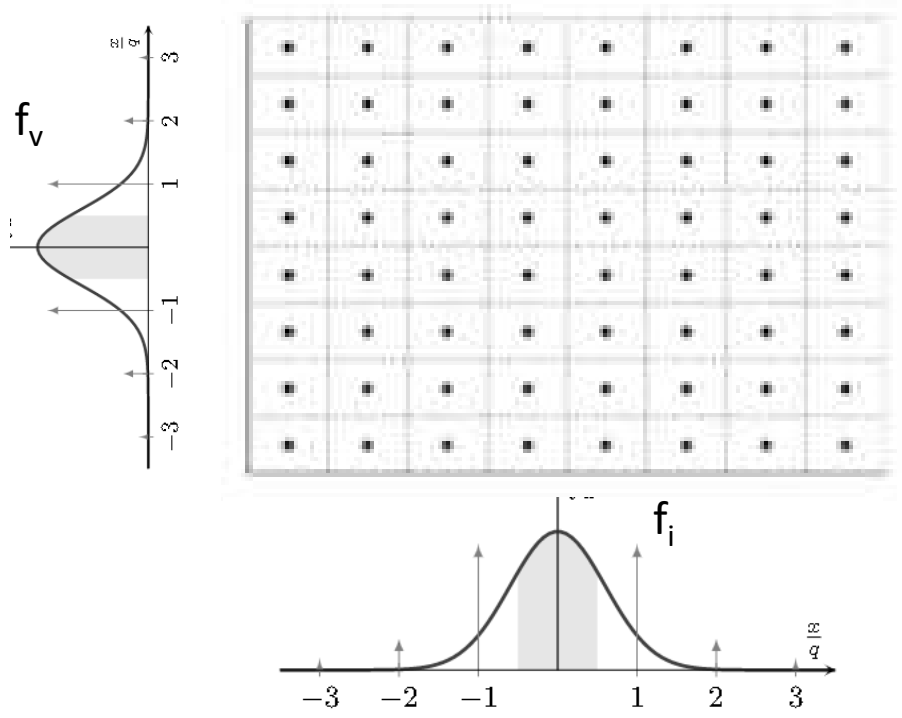
Imaginem que vamos quantizar duas variáveis

- V e I são independentes
  - Se usar quantizador uniforme pode-se assumir que teríamos uma divisão em matriz quadrada:



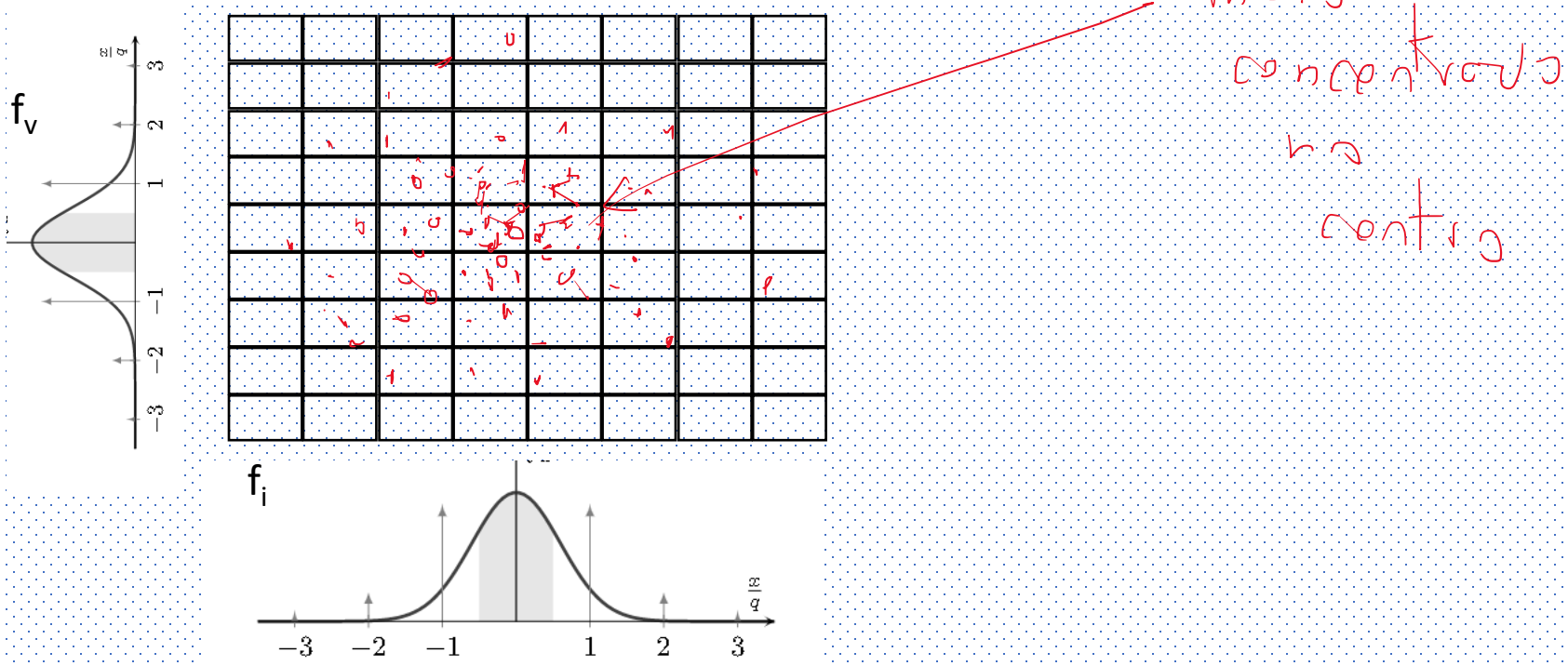
Imaginem que vamos quantizar duas variáveis

- V e I são independentes
  - Imaginando mesmo que suas distribuições não forem uniformes...



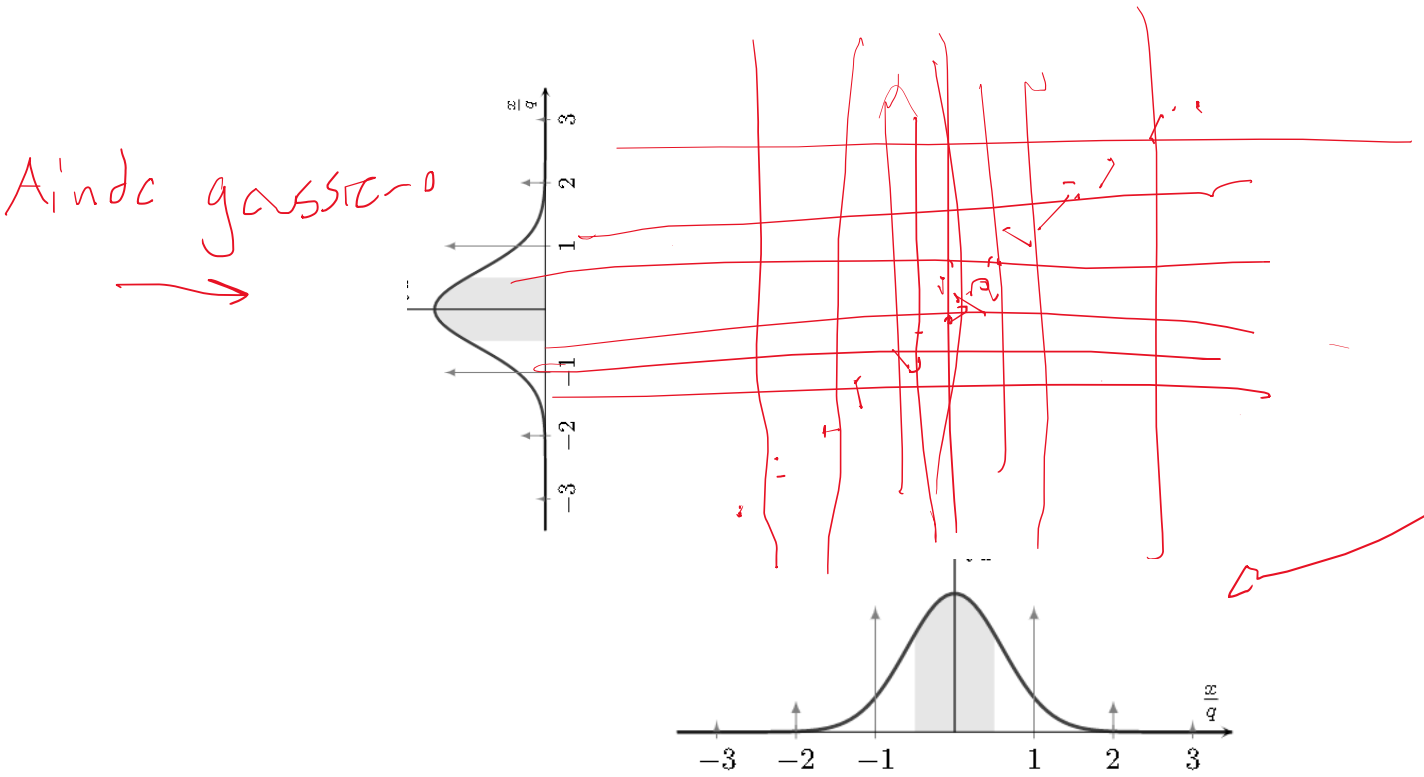
Imaginem que vamos quantizar duas variáveis

- V e I são independentes
  - Imaginando mesmo que suas distribuições não forem uniformes...



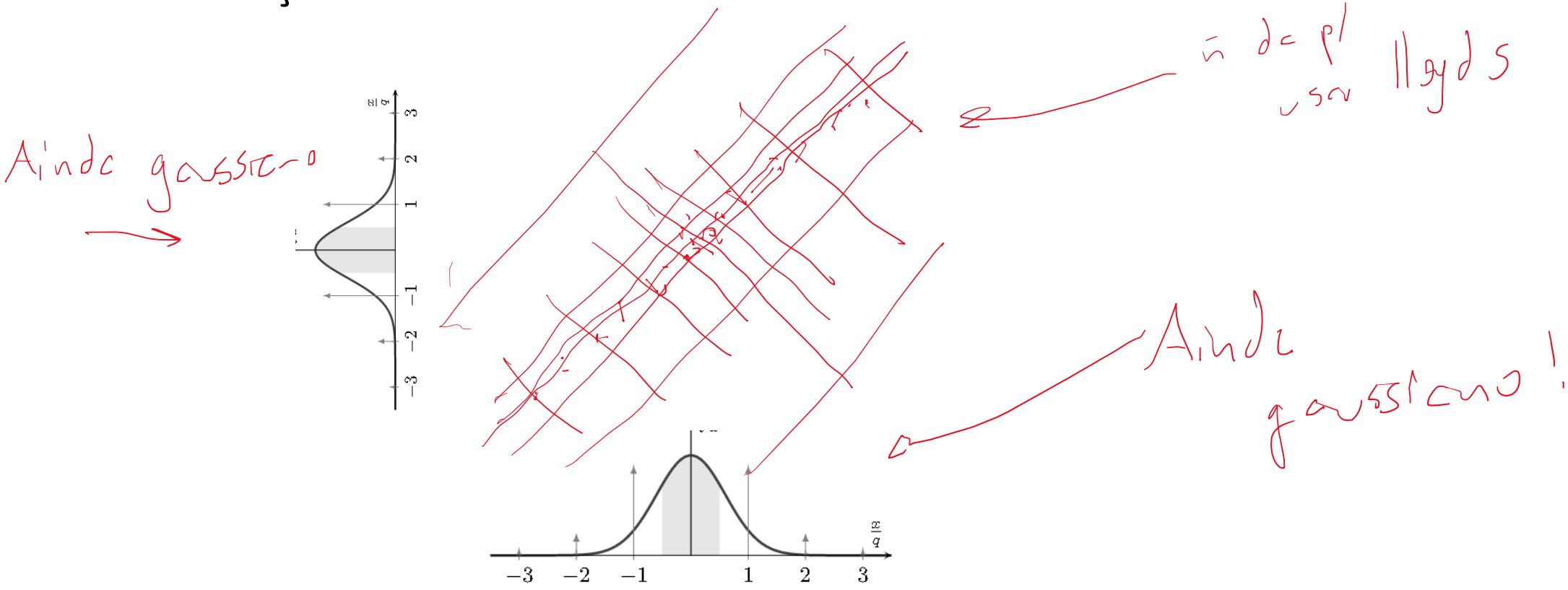
Imaginem que vamos quantizar duas variáveis

- Se V e I são dependentes
  - As distribuições de V e I não contam toda a história



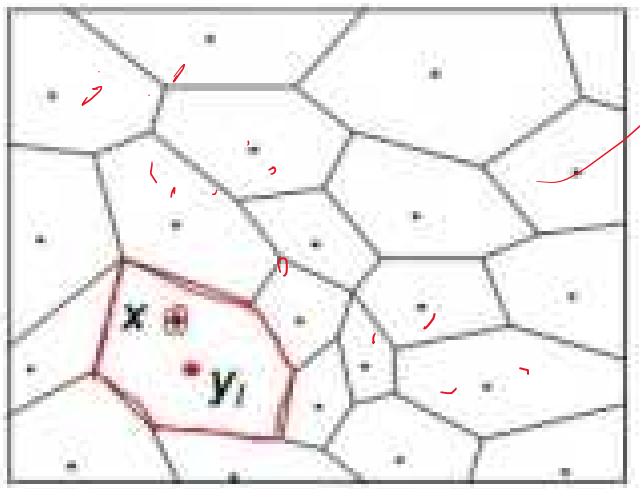
Imaginem que vamos quantizar duas variáveis

- Se V e I são dependentes
  - As distribuições de V e I não contam toda a história



Imaginem que vamos quantizar duas variáveis

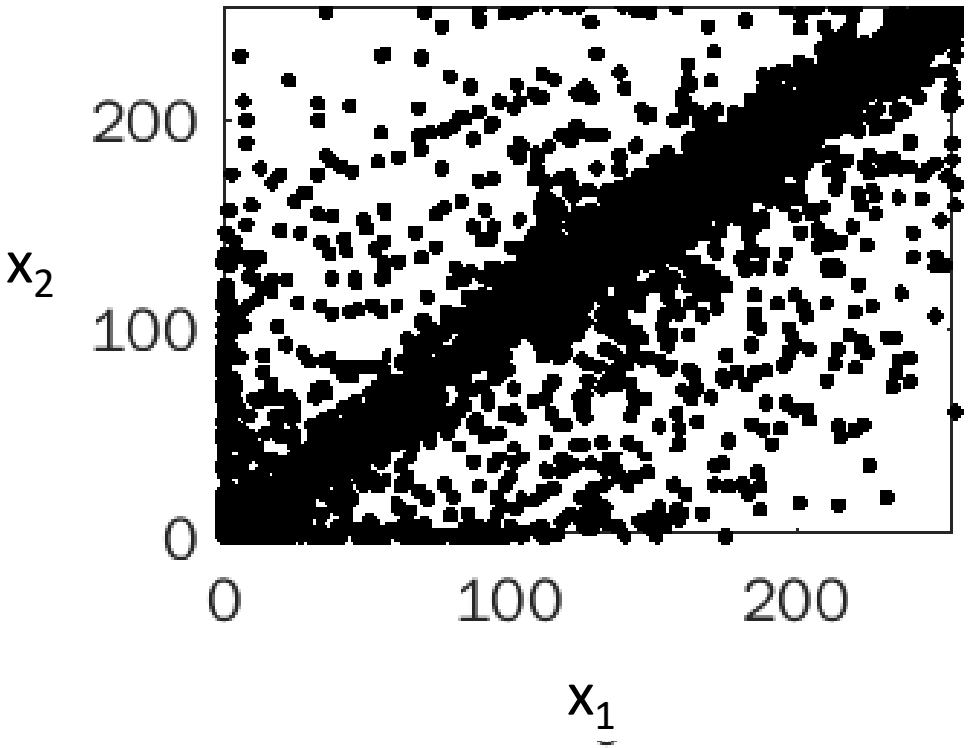
- Se V e I são dependentes
  - As distribuições de V e I não contam toda a história



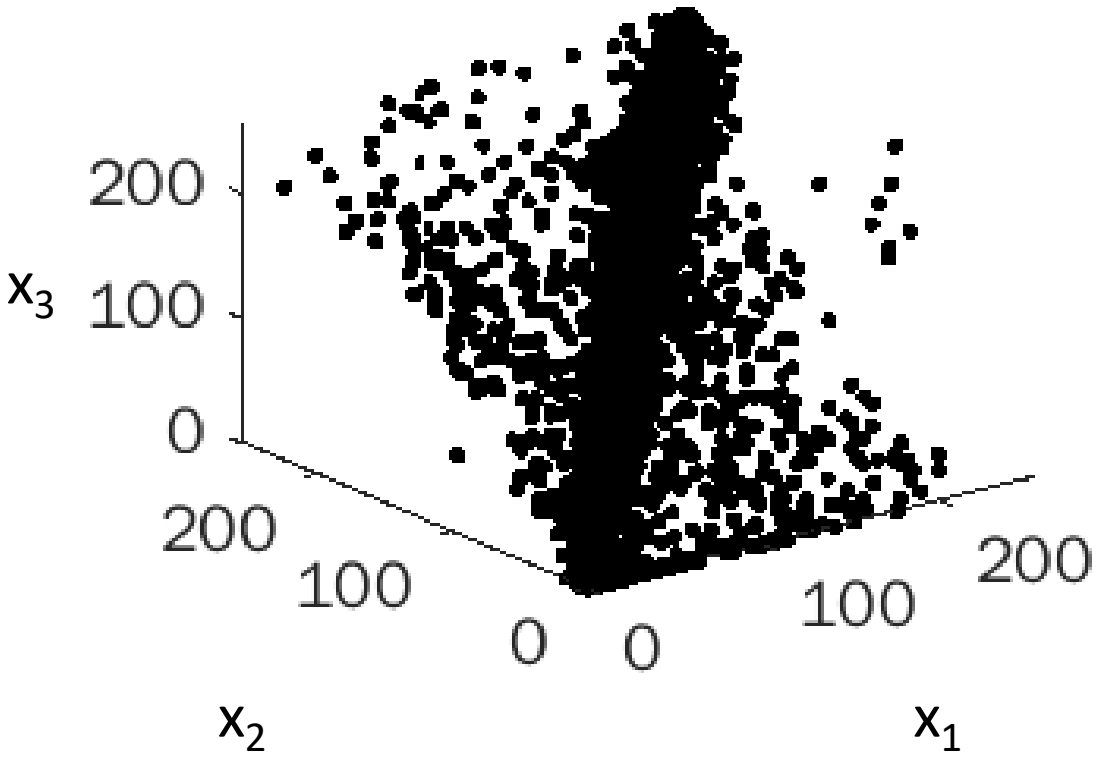
SIMULAÇÃO MATLAB

- No exemplo abaixo se analisa a imagem3 (grayscale), agrupando os seus coeficientes primeiro em vetores de dois e depois em vetores de três.

Tomando cada dois pontos subsequentes (impares - x_1, e pares - x_2) e fazendo v[x_1 x_2]

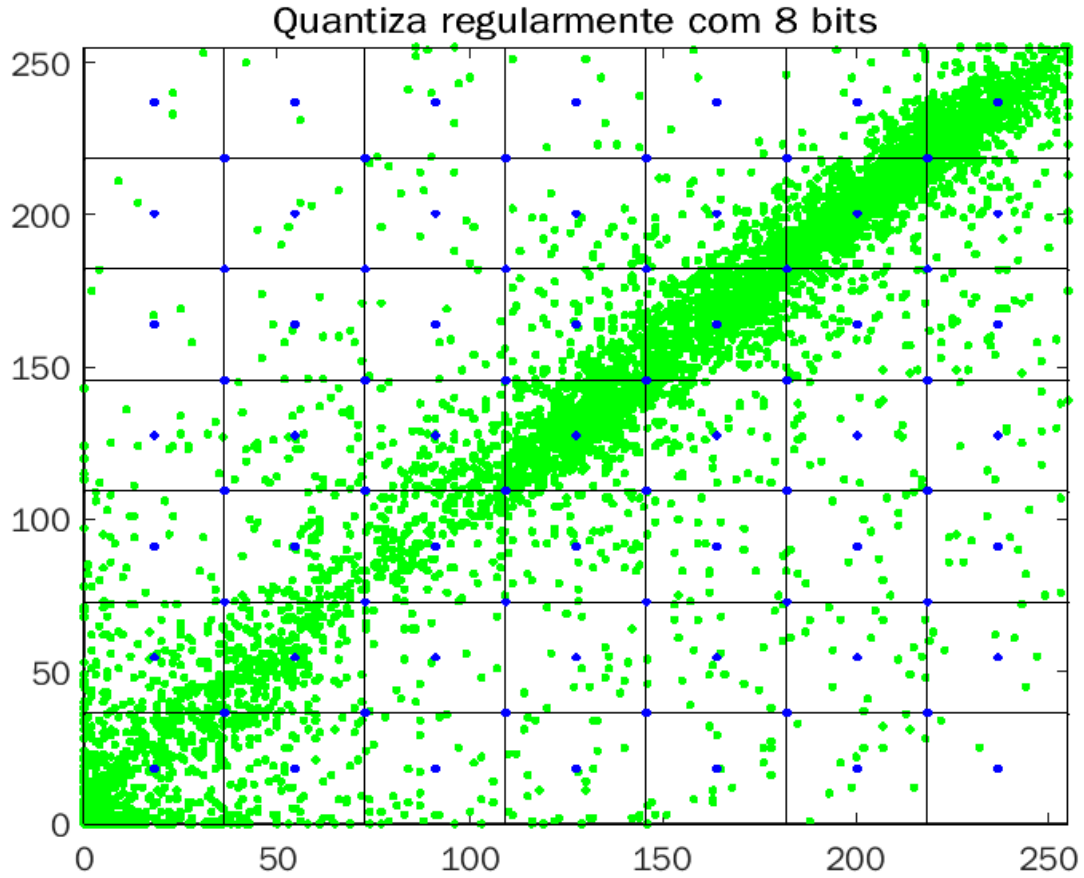


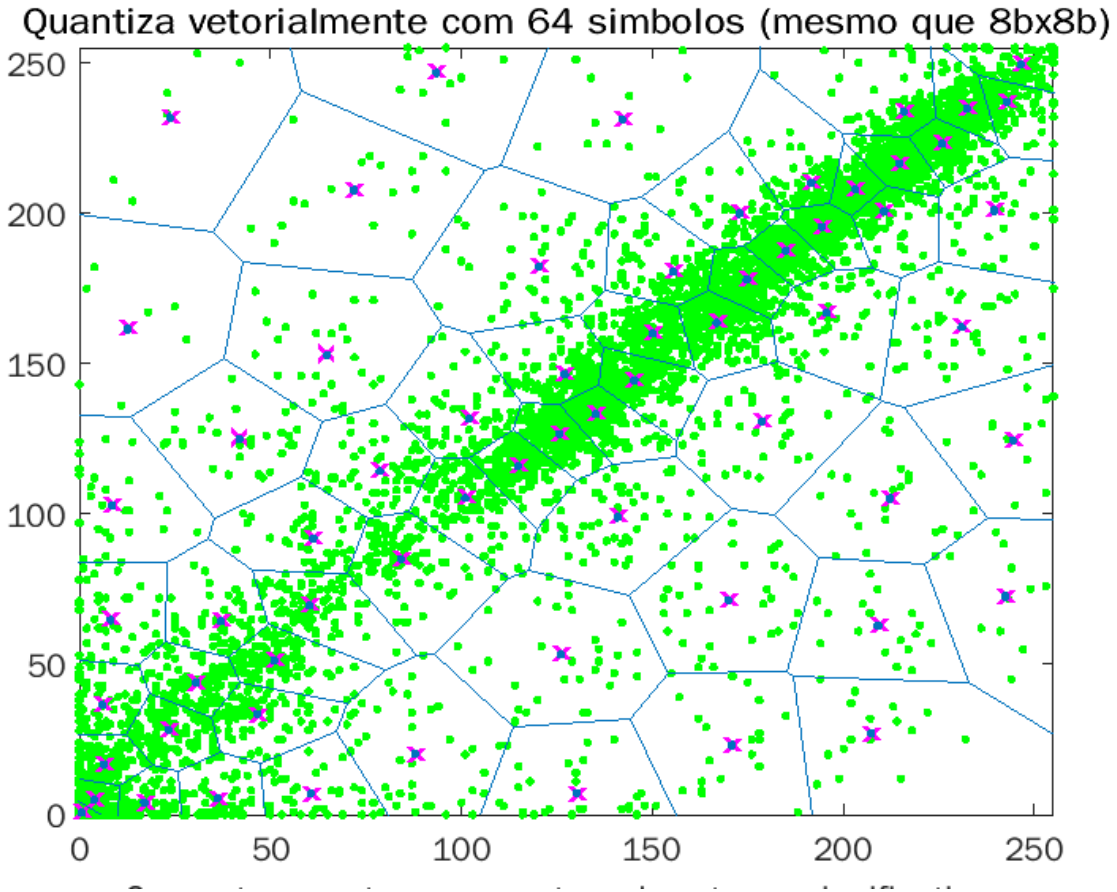
Repetindo, agora tomando os pontos três a três fazendo v[x_1 x_2 x_3]



Clusterização

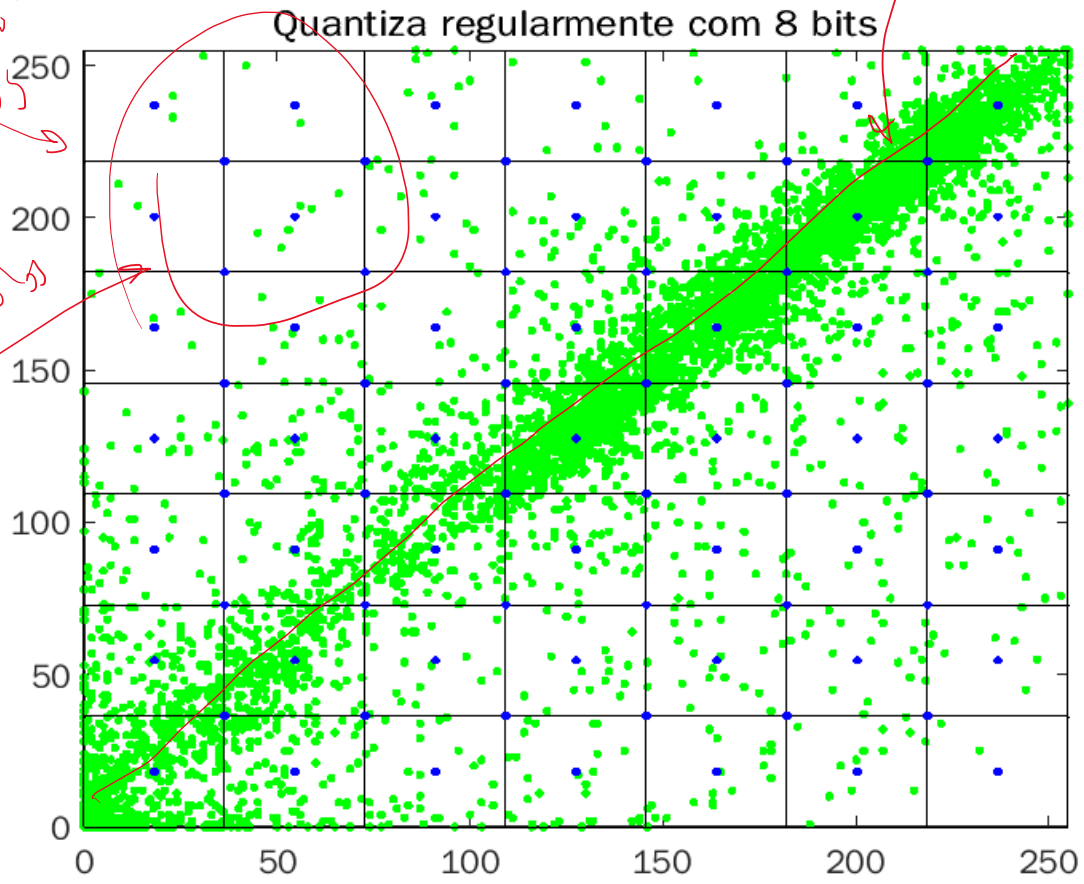
- Se usarmos a divisão regular de cada variável isso seria (por exemplo com r = 8 bits) isso é o mesmo que usar 64 símbolos.

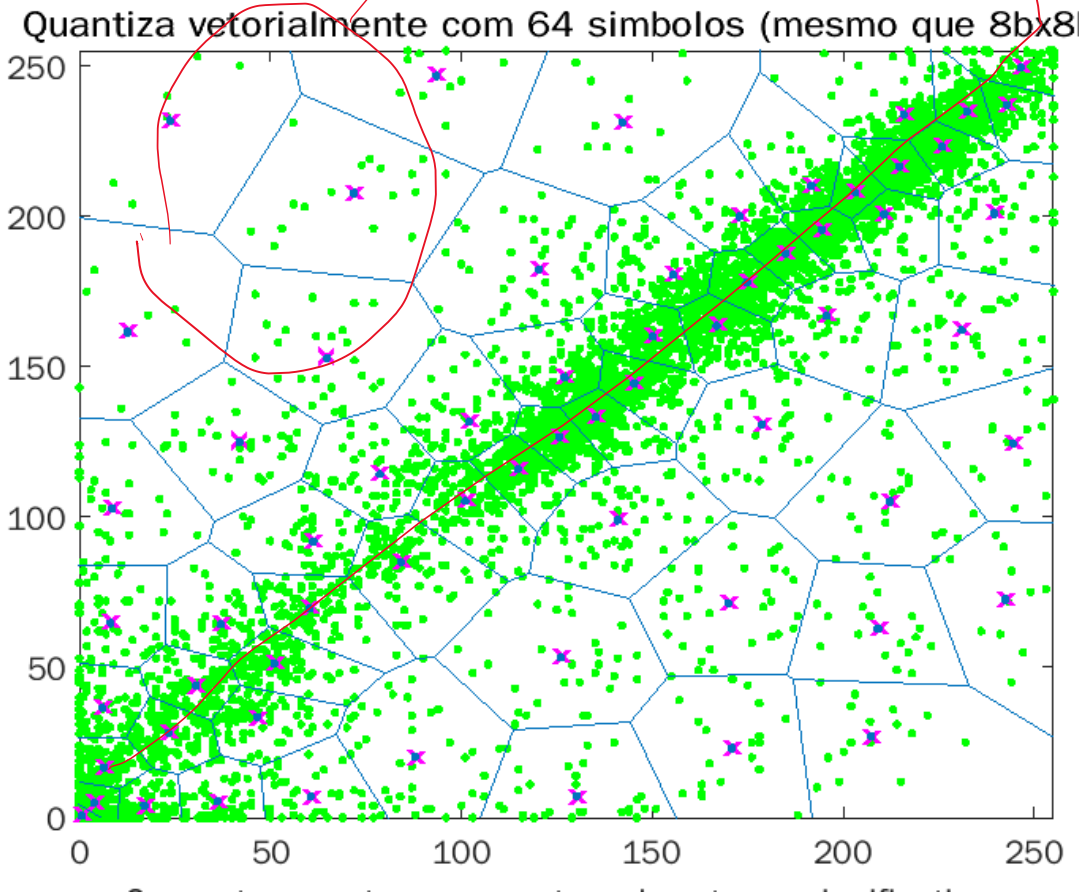




Clusterização

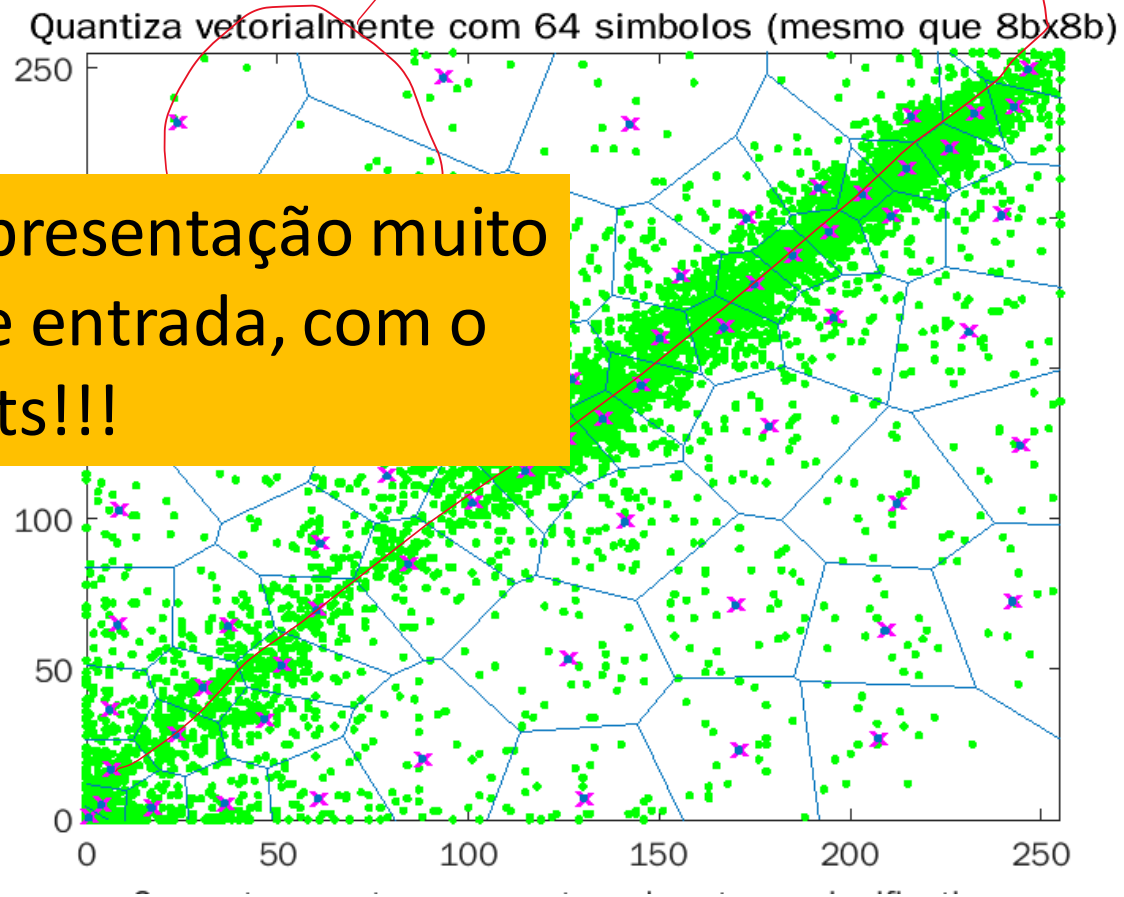
- Se usarmos a divisão regular de cada variável isso seria (por exemplo com r = 8 bits) isso é o mesmo que usar 64 símbolos.





Clusterização

- Se usarmos a divisão regular de cada variável isso seria (por exemplo com r = 8 bits) isso é o mesmo que usar 64 símbolos.



O resultado é uma representação muito melhor dos valores de entrada, com o mesmo número de bits!!!

Quantização vetorial (cont.)

- Agrupa-se a variável de entrada em vetores de N valores
- Utilizando a clusterização do espaço de variáveis de entrada se definem os centros que vão ser o novo codebook
  - Classifica-se as entradas sempre de acordo com sua proximidade dos centros definidos
  - Armazena apenas os índices para o codebook
- Na reconstrução
  - Com os índices define os vetores centro que representam cada conjunto de N entradas (cada vetor original)
  - Desagrupa-se os vetores reconstruindo os dados de entrada agora a partir dos centros definidos no codebook
- Como os valores serão mais próximos dos valores reais um mesmo número de simbolos (logo resolução) irá reconstruir o sinal com distorção bem menor.

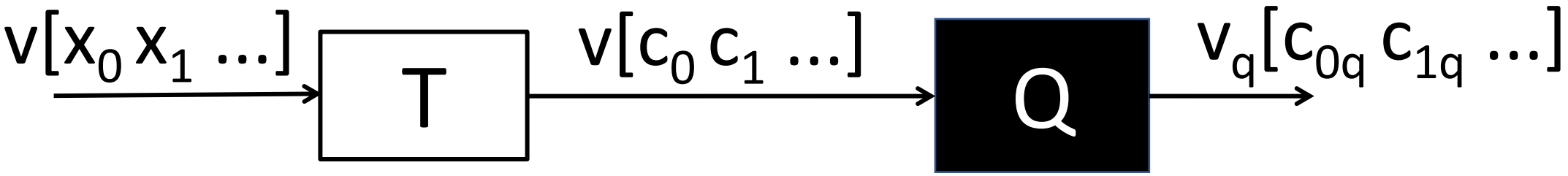
Introduz Transformadas

Transformadas em compressão

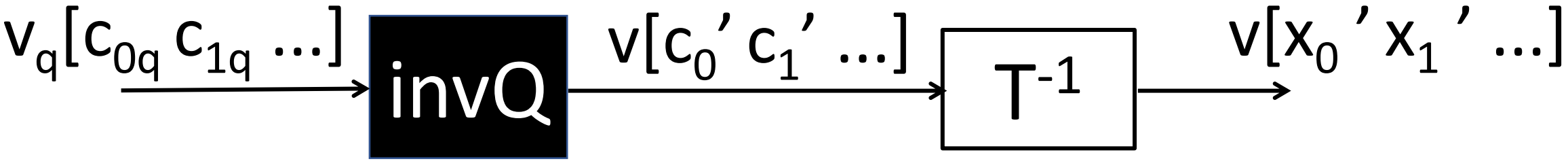
- Como vimos antes em muitas situações quando se tem um conjunto de dados para analisar esses podem ser agrupados em um bloco ou vetor.
- Quando se analisa o vetor em si esse não é um vetor que, ao longo do conjunto universo de dados ocupa seu espaço de forma compacta.
- De fato, a justificativa para usar quantização vetorial é essa.
- Mas se poderia proceder de forma alternativa procurando aplicar uma transformação T aos vetores de entrada a fim de reorganizar os dados de forma que a quantização posterior tivesse ganhos.

Aplicação de Transformadas

- A idéia aqui é aplicar a transformada T de forma que a Quantização subsequente seja mais efetiva na compactação dos dados:



- Na decodificação após o reescalonamento (dequantização) o sinal passa por uma transformação inversa e recupera uma aproximação dos dados



Transformadas

- Imaginando que nossos vetores (ou blocos) de entrada descrevem a representação de um conjunto de dados que não é totalmente aleatório (compacto), assume-se que existe uma outra base na qual esses seriam melhor representados.
- Essa outra base pode ser pensada como uma transformação do espaço vetorial inicial, ou, alternativamente, como uma nova base no espaço de funções que melhor representa uma função definida por esse conjunto de dados.

Exemplos de transformadas usadas em compressão

- Diversas são as transformadas que podem ser utilizadas para a compressão de dados:
  - Discrete Cosine Transform, Discrete Fourier Transform, Wavelet Transforms (many types and variations), Walsh Hadamard Transform, e Karhunen Loeve Transform.
  - Aplicações específicas podem trazer ainda ideias bastante variadas como:
    - Haar-Hadamard Transform (Haar é um tipo de wavelet) e
    - Empirical Mode Decomposition (um dos elementos da chamadas transformada de Hilbert Huang), para compressão de imagem;
    - Discrete Hilbert Transform e FFT para a compressão de características de ECG ...
- Futuramente falaremos em detalhe sobre várias dessas transformadas

A Transformada de Karhunen-Lóeve

- Uma boa introdução ao conceito, por ter uma ação de fácil entendimento para quem já viu a quantização vetorial é a transformada de Karhunen-Loéve (KLT)
- A mesma ideia por trás dessa transformação é encontrada em diversas formas na literatura técnica:
  - Transformada de Karhunen-Loéve (KLT)
  - Análise de Componente Principal (PCA)
  - Single Value Decomposition (SVD)
  - Decomposição de Karhunen-Lóeve
- O conceito é de buscar uma base orthogonal do conjunto de dados e usa-la na decomposição desses dados.

Qual a dimensão “certa” para um conjunto de dados não ser redundante

- Pensando no nosso problema do agrupamento de dados em blocos ou vetores (blocos sempre podem ser vetorizados)
- Existe uma dimensão de vetor que é a mais indicada para representar meus dados?
  - Se existe, como descobrir essa dimensão (pensem que em conjuntos com N dimensões – N>3, uma visualização dos dados não é possível)?
  - Expressando de outra forma: Como encontrar padrões e agrupamentos em um espaço de dimensão elevado?
- Esse é um problema que é tão recorrente em todos os campos de pesquisa que a KLT foi redescoberta várias vezes de forma independente (com ligeiras variações de aplicação).

Matriz de covariância

- Imaginando que os dados tenham N dimensões se pode verificar o quanto cada uma dessas dimensões varia em relação a si própria (variância) e a cada uma das outras (covariância)
- Organizando esses dados de forma que $\sigma_{ij}^2 = E\{c_i \cdot c_j\}$.
- Na prática se aproxima o valor esperado sobre o conjunto de finite de L vetores de N dimensões que compreende o nosso conjunto de dados. $\sigma_{ij}^2 \sim \frac{1}{L}\sum_{n=1}^{L}\left(c_{n,i} - \overline{c_{n,i}}\right) \cdot \left(c_{n,j} - \overline{c_{n,j}}\right)$
- Como $\sigma_{ij}^2 = \sigma_{ji}^2$, a matriz é simetrica em relação a diagonal e quadrada com dimensões N x N

$$C = cov\left(c_{1:L,1:N}\right) = \begin{vmatrix} \sigma_{11}^2 & \cdots & \sigma_{1N}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{1N}^2 & \cdots & \sigma_{11}^2 \end{vmatrix}$$

Autovalores e Autovetores

- Para matrizes quadradas C pode-se encontrar os autovetores v e autovalores λ que obedecem a relação:

$$M \cdot v = \lambda \cdot v$$

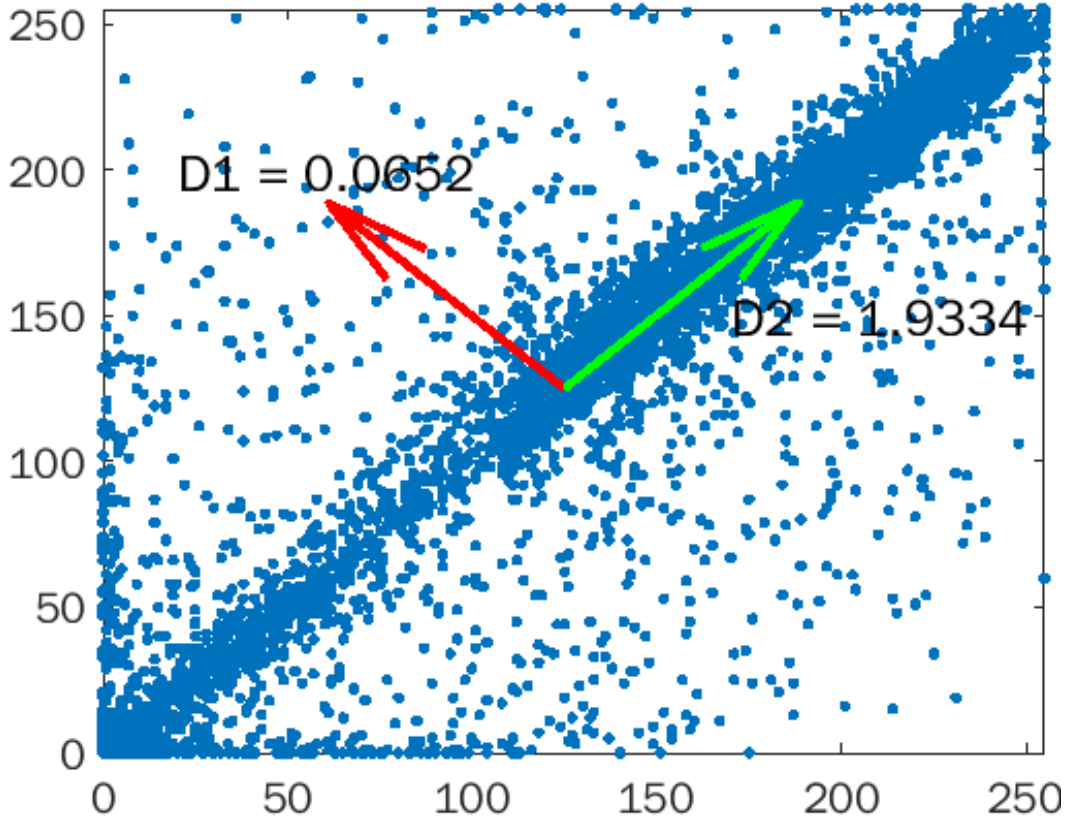
- Se os autovalores e autovetores de uma matriz quadrada de NxN existir serão N autovalores e N autovetores
- Os autovetores são ortogonais entre si, portanto podem formar uma base

A análise de componente principal

- Os autovetores da matriz de covariância indicam as direções em que o conjunto de dados varia.
- Já os autovalores indicam o peso relativo de cada uma dessas direções: o peso do vetor verde é quase trinta vezes maior que o vetor vermelho

$$D2 = 29.64 \times D1.$$

- O peso está sendo dado pelo autovalor normalizado pelo número de vetores.

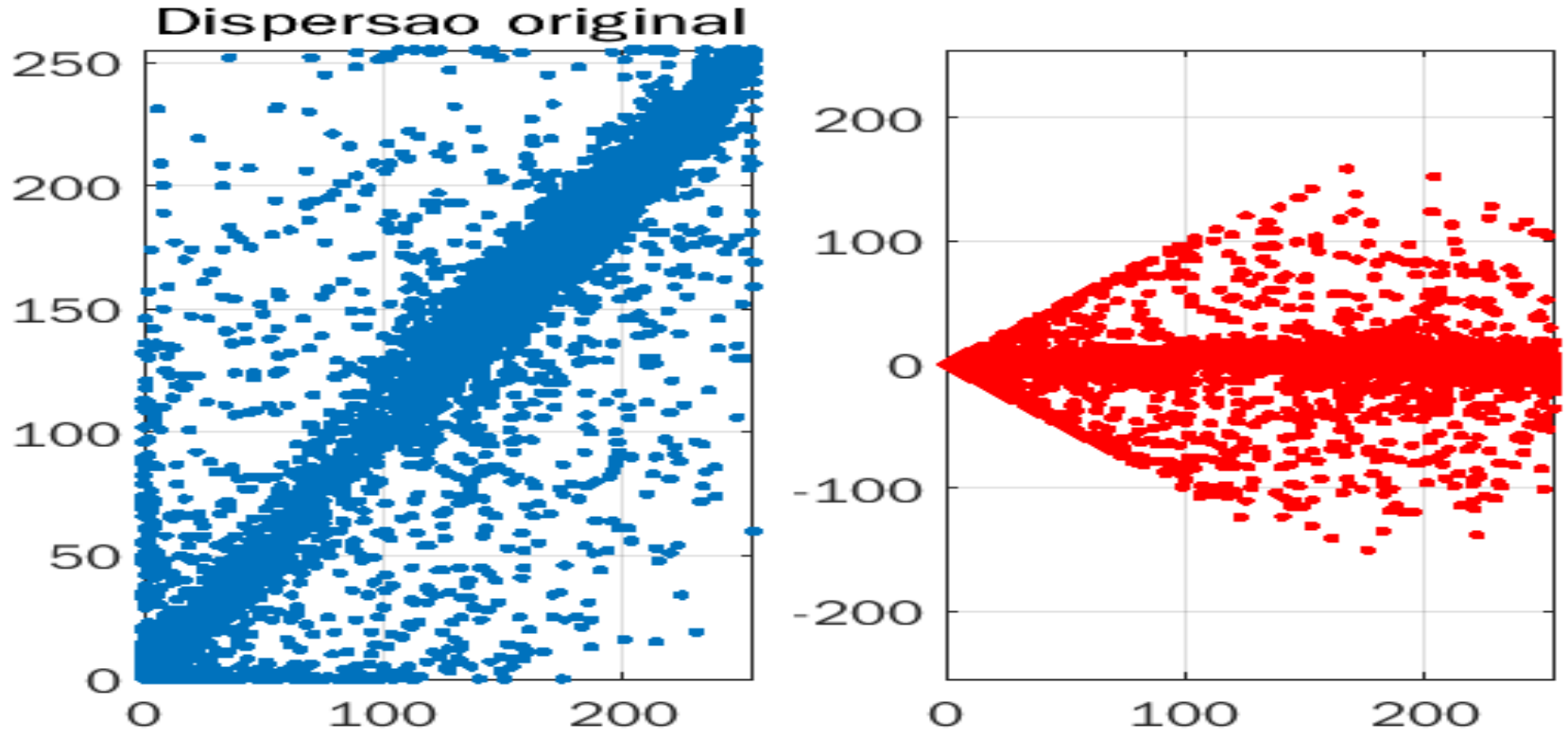


A analise de componente principal

- A idéia por trás da análise de componente principal é que os eixos em que o peso é muito baixo eventualmente podem ser desconsiderados (em especial em espaços com N grande).
- Para isso, a primeira coisa que se deve fazer é mudar a base dos pontos de forma a alinhar eles com a decomposição obtida.
- Isso pode ser feito multiplicando a matriz de vetores base V pelos dados de entrada C: $C_{KLT} = v^T \cdot C^T$
- A transformada inversa é obtida repetindo-se a operação.
- Esse processo é a transformada KLT.

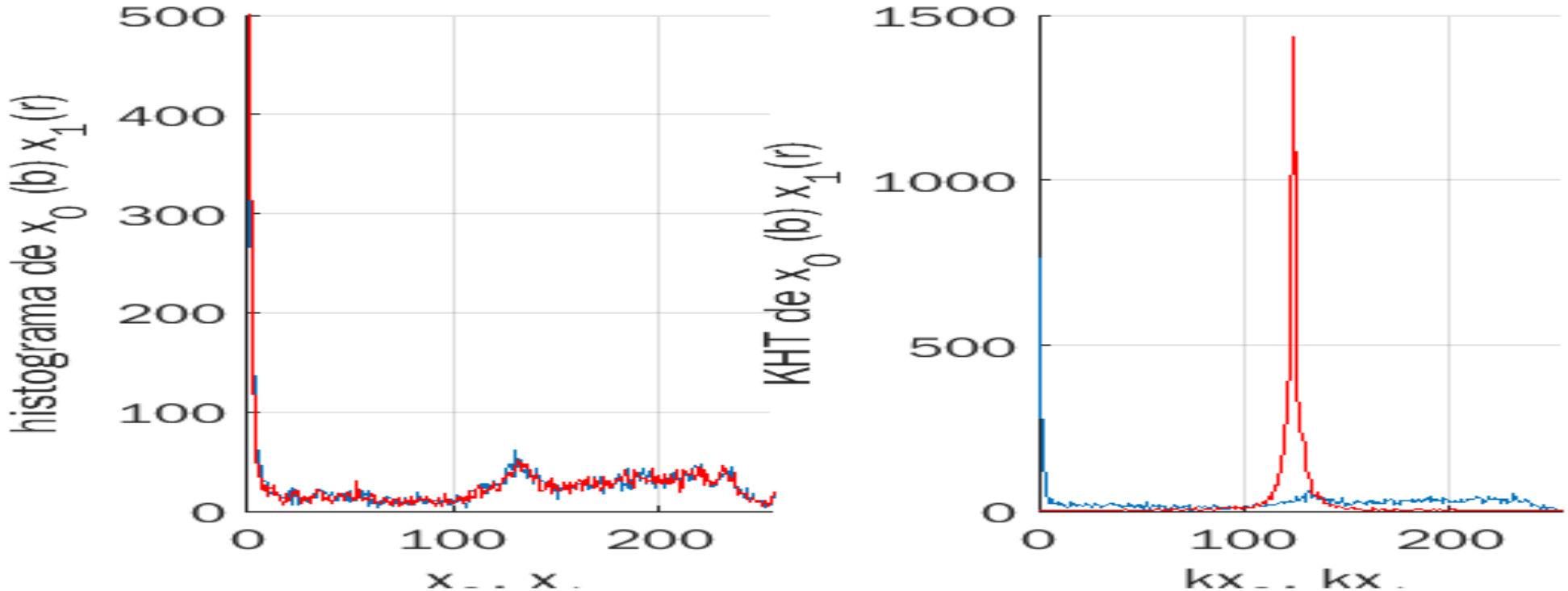
MATLAB

- Na figura abaixo se mostra o efeito da KLT sobre o conjunto de dados da imagem observada anteriormente



MATLAB (cont)

- Observando os histogramas é possível notar que uma quantização posterior reduziria dramaticamente o espaço ocupado pelos valores no eixo vermelho pois seus valores variam muito menos (eixo 1).



Na próxima aula

- Finalizar o uso de transformadas em compressão
  - Transformadas para imagem e vídeo
  - Transformadas para som e outros sinais monodimensionais.

- Colocar questões de predição e quantização no tempo para sinais monodimensionais
  - Quantização delta e delta sigma
  - LPC