



# Processamento Digital de Imagens

## Conceitos Básicos

E Carto/UFPR

Prof. Dr. Jorge Centeno



# Realidade e imagem

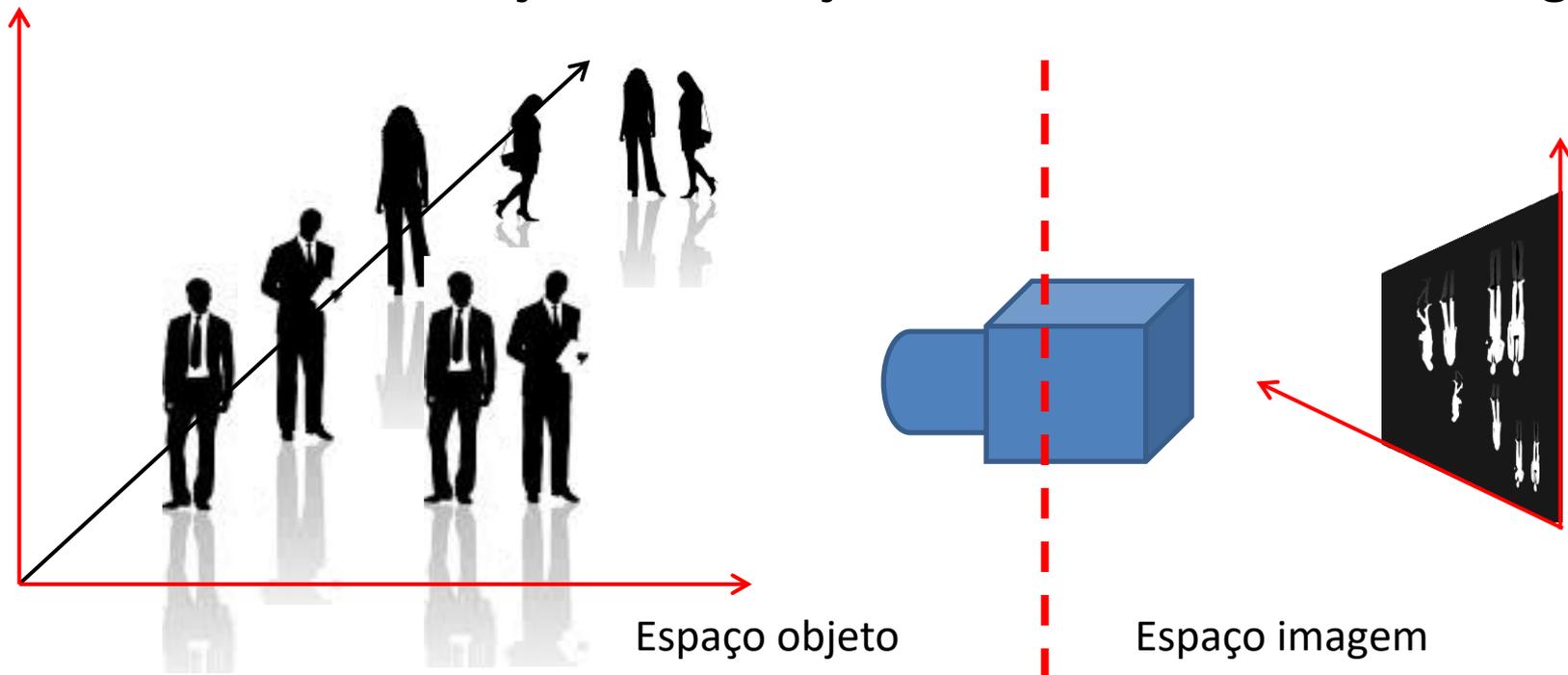
- Uma imagem é a representação pictórica de um aspecto da realidade.
- Uma imagem não é idêntica à cena real, pois ela retrata apenas uma propriedade particular:
  - Radiação refletida
  - Radiação transmitida
  - Etc
- deixando de registrar algumas propriedades e relações que podem ser importantes para a adequada compreensão da cena.



# A considerar

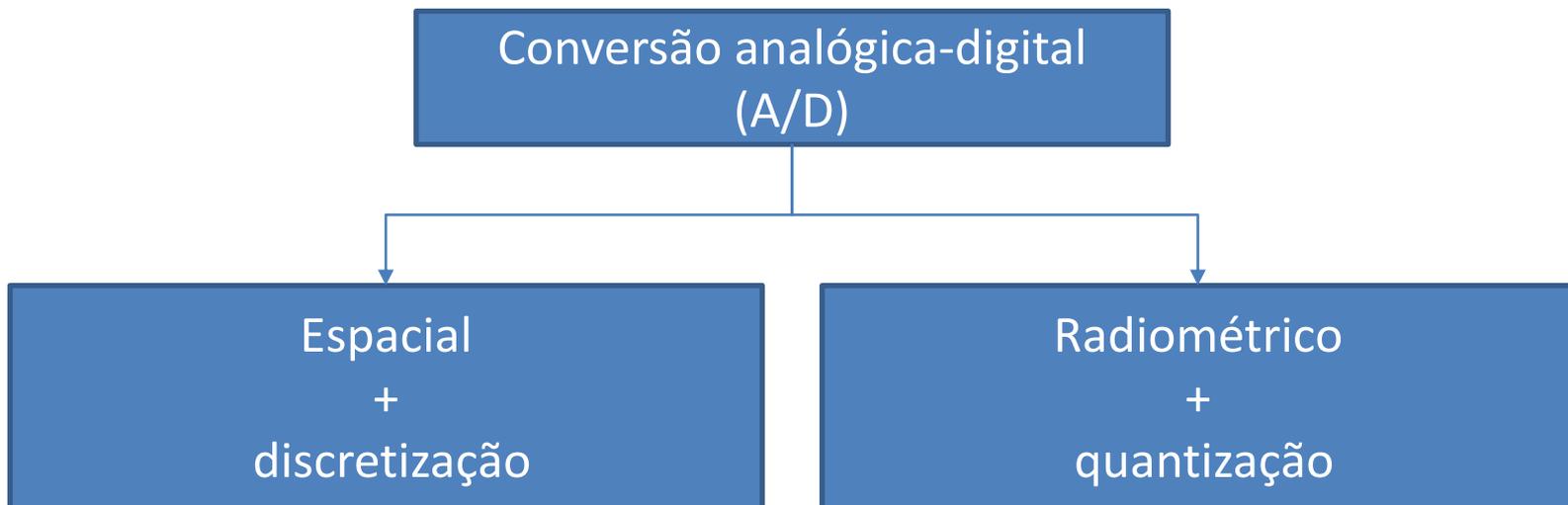
Transformação espacial: o espaço objeto e o espaço da imagem.

- O registro da radiação incidente no sensor
- A transformação da radiação incidente em um valor digital



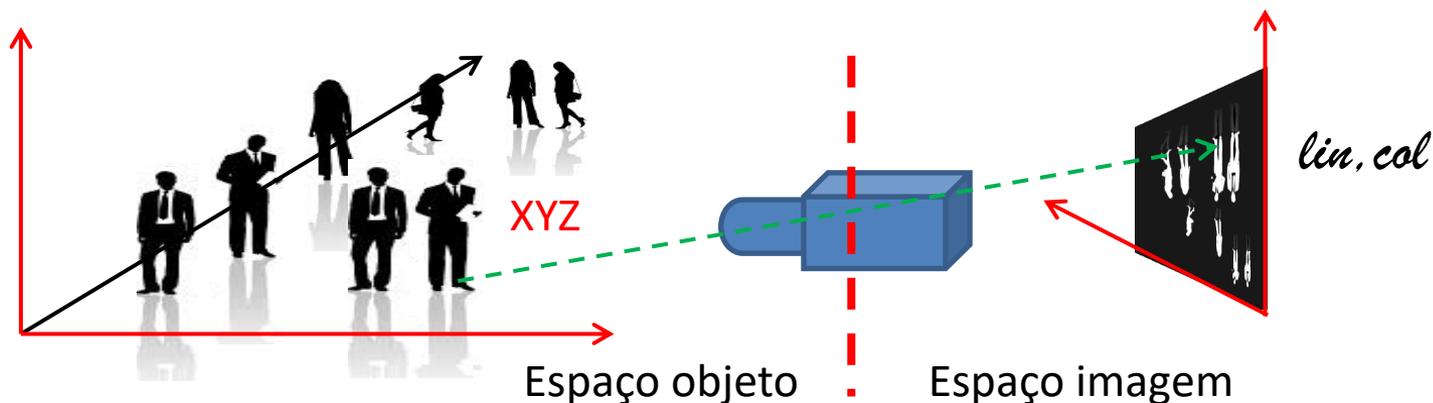
# A formação da imagem

No processo de registro da radiação incidente no sensor, a cena é simplificada de várias maneiras. Os principais aspectos são o Aspecto Geométrico e o Radiométrico, que envolvem dois conceitos, a discretização e a quantização.

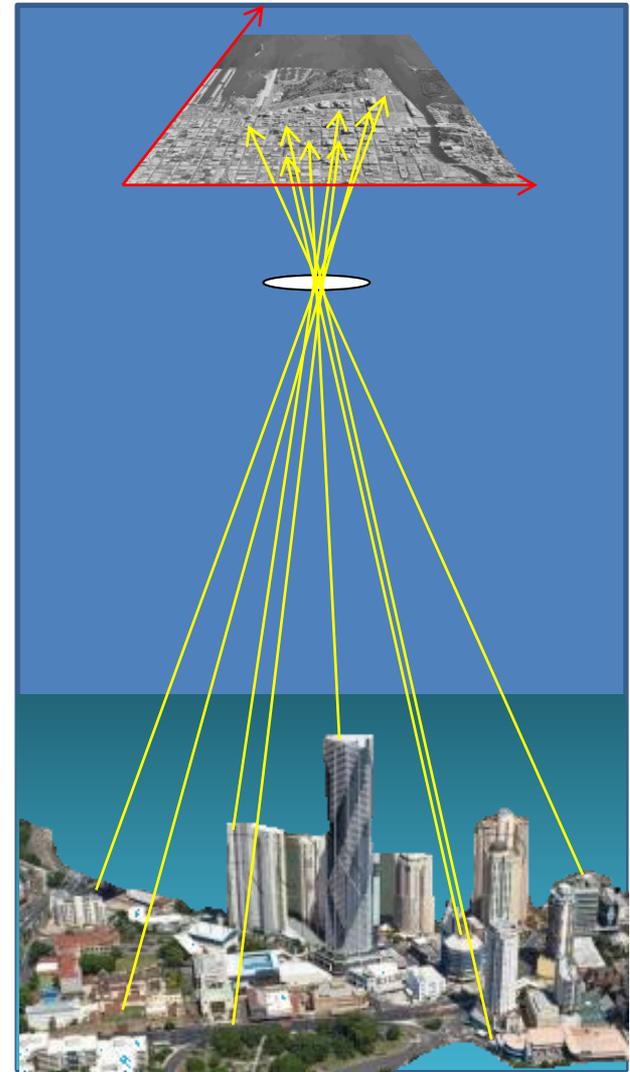


# O aspecto geométrico

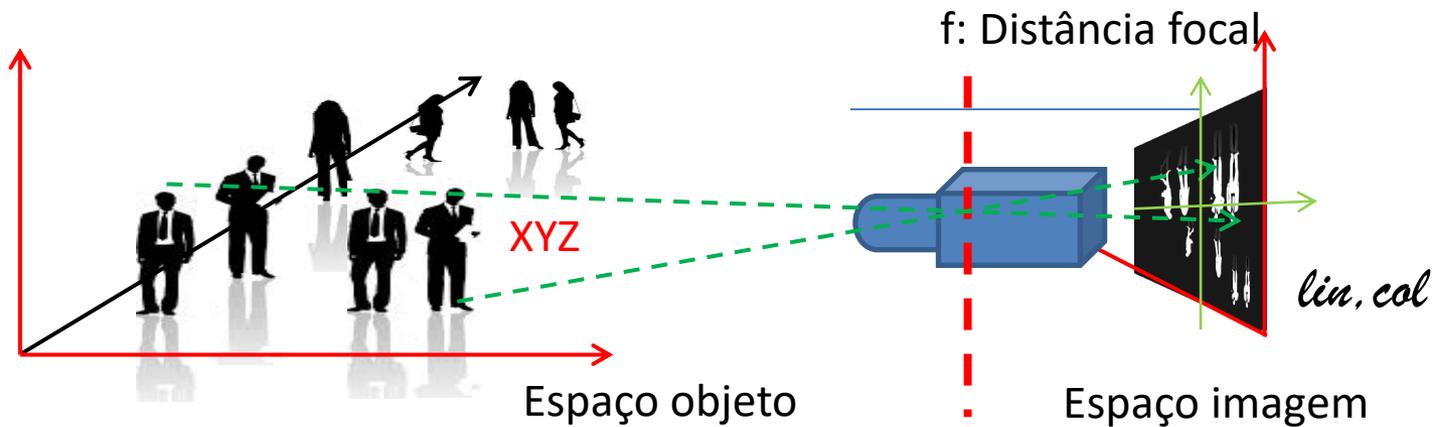
- O aspecto geométrico determina a posição dos objetos (pontos) existentes na cena (mundo real) no espaço da imagem. Ele responde à pergunta:
- *“Onde se encontra o objeto, com coordenadas conhecidas no mundo real ( $X; Y, Z$ ) dentro do plano da imagem?”*
- Ou...
- *“A que ponto no espaço objeto corresponde o ponto com coordenadas (linha, coluna) na imagem?”*



- Basicamente, a coleta de imagens é o resultado da projeção de um espaço tridimensional (espaço objeto) num espaço bidimensional (espaço imagem).
- Esta transformação envolve perda de informação espacial, pois uma coordenada é perdida.
- A formação da imagem podem ser modelada usando a transformação da projeção central. É assumido que todos os feixes de luz passam pelo mesmo ponto no plano focal e incidem no plano da imagem localizado a uma distância  $d'$  do plano focal, com as coordenadas:
  - $x = (x_1, x_2, d')$



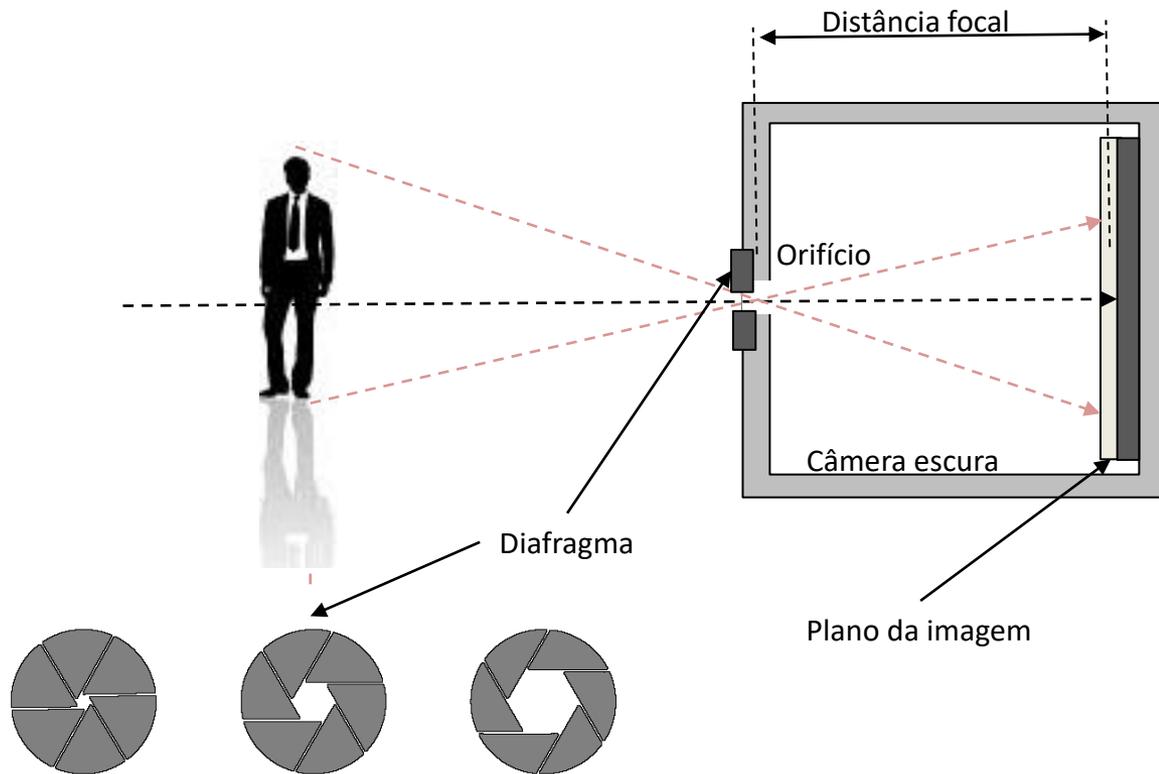
# Pinhole model simplificado



$$(lin, col) = F(X, Y, C, f)$$

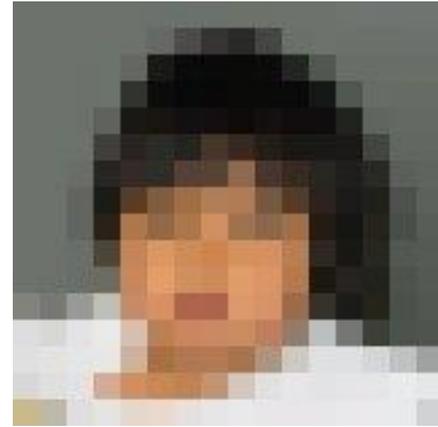
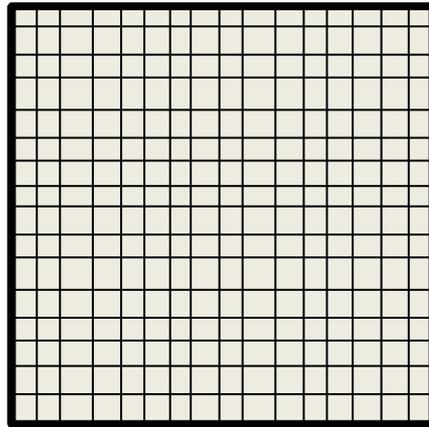
- Conseqüências:
- Vários objetos do espaço tridimensional podem ser projetados no mesmo ponto da imagem: Oclusão.
- O modelo de câmara “pinhole” é uma simplificação. Para ser mais fiel à realidade e modelar adequadamente esta relação espacial, é necessário levar em consideração outros fatores como as deformações introduzidas pelo sistema óptico ou os efeitos da atmosfera na propagação do feixe de luz.

- Modelo pin-hole

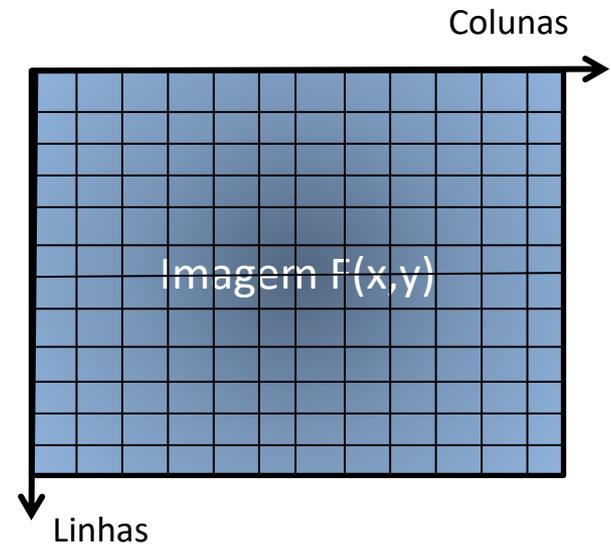


# Discretização

- Processo de simplificação do plano da imagem sob forma de uma malha regular finita, cujos elementos têm um tamanho fixo e constante (pixels).
- Assim o plano da imagem é representado sob forma de uma matriz. A radiação incidente em cada célula é medida por um elemento detector.

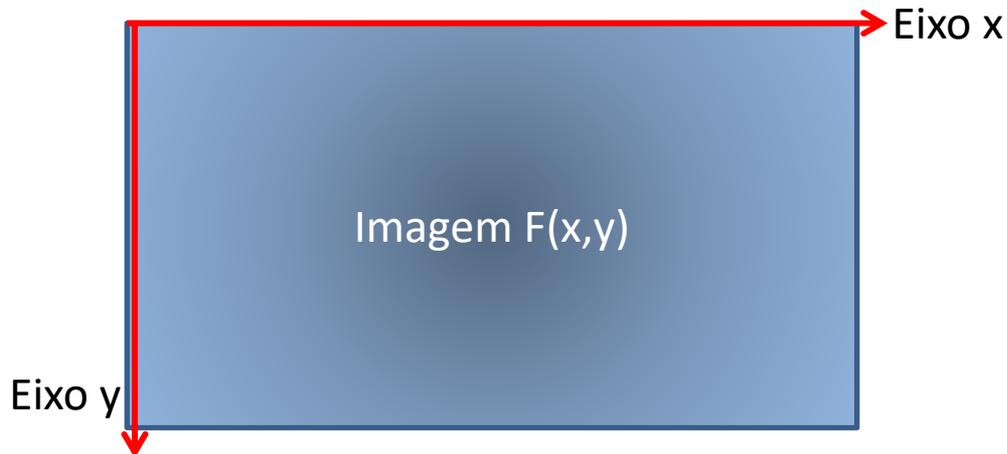


- Com a discretização as possíveis posições dos pontos no espaço definido pelo plano da imagem são restritas a valores inteiros dentro da faixa de variação definida pelo tamanho da grade (NxM)
  - $F(x,y)$  com  $x=1,2,3,\dots,N$  e  $y=1,2,3,\dots,M$
- Sendo assim, cada elemento detector registra a energia incidente em uma unidade de área do plano da imagem, do que se deriva um valor único a ser armazenado na matriz digital.



# Aspectos radiométrico

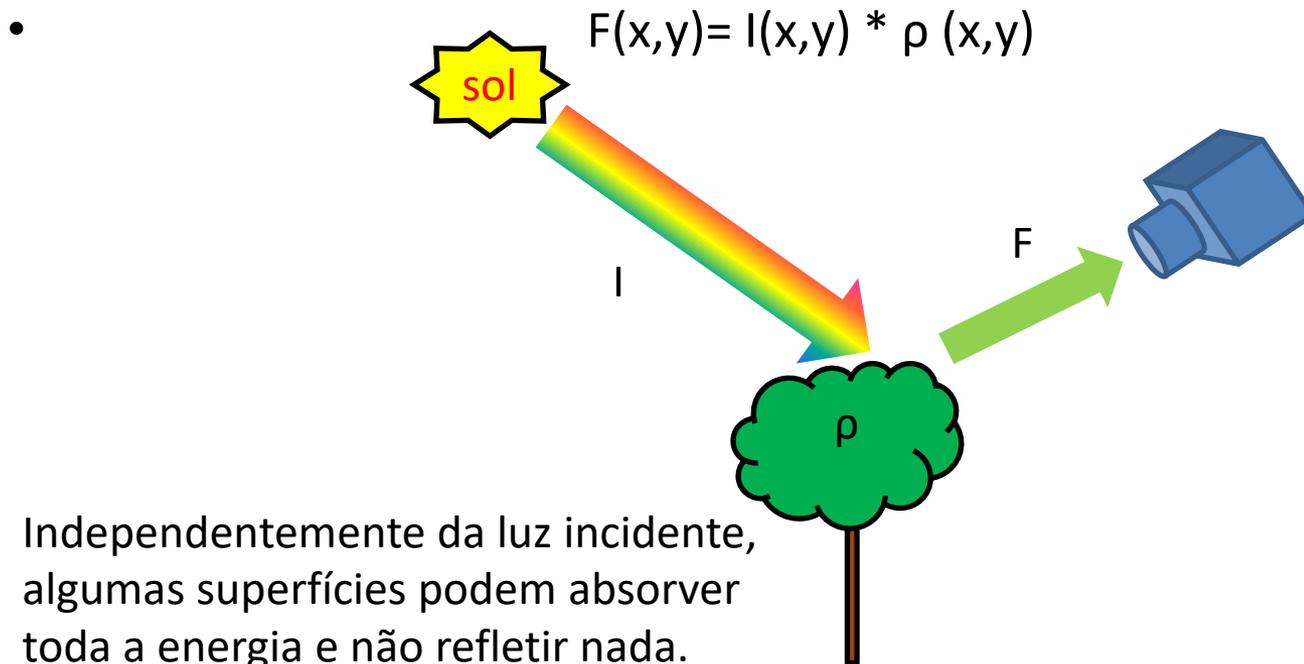
- Está associado à maneira como o brilho ou a energia proveniente de um objeto ou superfície é registrado e a fidelidade com a que ele é representado na imagem.
- O valor correspondente a cada pixel é definido em função de sua posição na imagem como:
  - »  $F(x,y)$
- Onde  $x$  e  $y$  representam as coordenadas no plano da imagem. Note a origem no canto superior esquerdo e o sentido dos eixos.



# Aspectos radiométrico

Em uma fotografia,  $F(x,y)$  se registra a radiação eletromagnética refletida pelos objetos. A energia incidente no sensor está associada a duas grandezas físicas:

- intensidade de energia incidente no **objeto** ( $I$ ) e
- **reflectância** do objeto ( $\rho$ ), ou sua capacidade de refletir a radiação incidente.



Independentemente da luz incidente, algumas superfícies podem absorver toda a energia e não refletir nada.

Como as imagens são obtidas em determinadas faixas espectrais, esta relação deveria ser melhor explicitada em função do comprimento de onda da energia utilizada para obter a imagem, ou seja, a faixa espectral à qual o sensor ou filme é sensível:

$$F_{\lambda}(x,y) = I_{\lambda}(x,y) * \rho_{\lambda}(x,y)$$

Com

$$0 \leq I < \infty$$

$$0 \leq \rho(x,y) \leq 1$$

A reflectância ( $\rho$ ) é a grandeza física que mede a proporção da energia incidente que o objeto é capaz de refletir num comprimento de onda determinado. Ela é uma propriedade intrínseca da superfície e depende da natureza e estado do objeto.



Red

Green

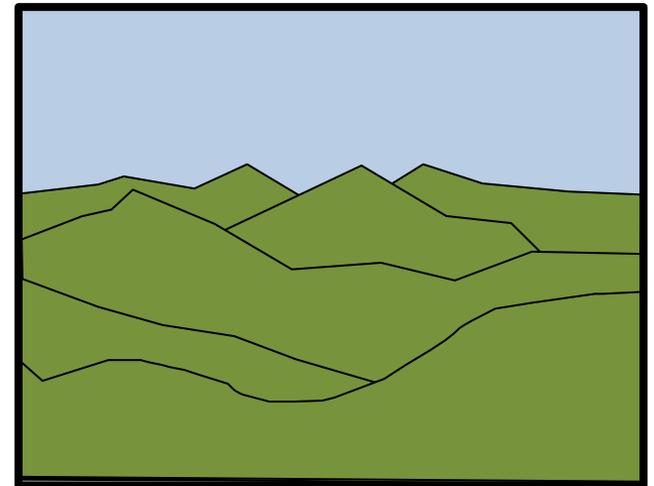
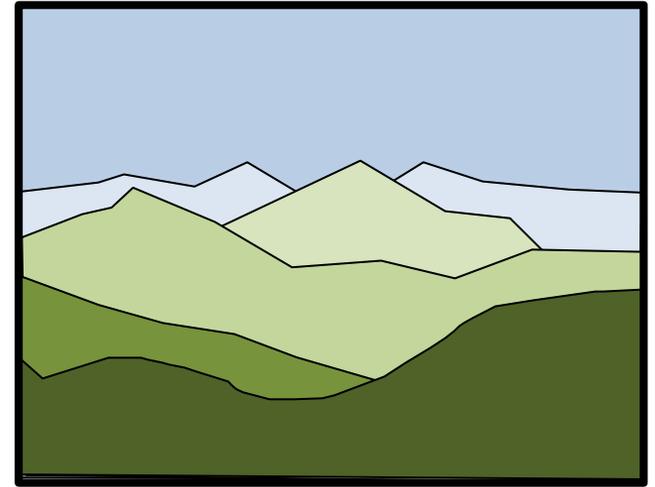
Blue



Outros fatores são responsáveis pela atenuação da energia, como por exemplo o efeito da atmosfera ou a opacidade do sistema de lentes utilizado para captar a imagem, uma função de atenuação (A).

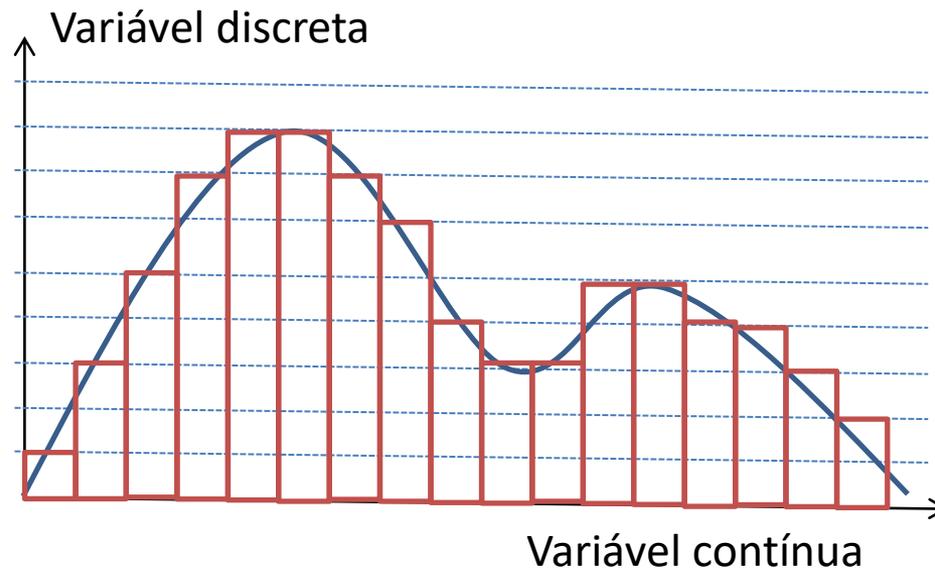
- $F(x,y) = A(x,y) * I(x,y) * \rho(x,y)$

É comum, encontrar os efeitos da atmosfera em fotografias aéreas tiradas de grande altitude, ou em imagens de satélite. Mas você também pode notar isto ao ver uma cadeia montanhosa à distância.



# Quantização

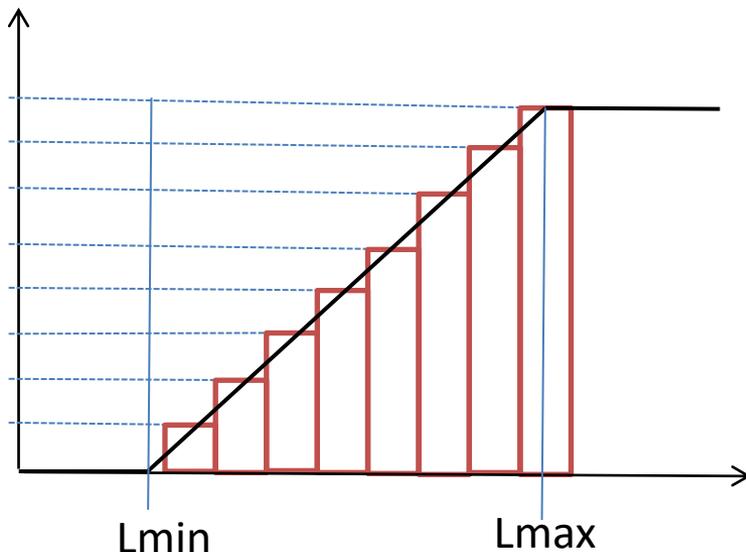
- A energia incidente no sensor é uma grandeza contínua. Sua completa representação (de forma digital) demandaria o uso de muita memória.
- Na prática costuma-se armazenar apenas uma quantidade finita, mas suficientemente grande, de níveis de energia em lugar de todos os valores possíveis. É feita então uma segunda discretização, desta vez nos valores armazenados nos pixels.



# Quantização

- Para isto, é definida uma escala discreta de valores, com valores  $L_{min}$  e  $L_{max}$  fixos e uma quantidade de valores possíveis.
- Esta quantidade é chamada de resolução radiométrica e, na prática, é definida pelo número de bits usados para armazenar a informação. Geralmente, são usados 8 bits para cada pixel, o que corresponde a um total de 256 diferentes níveis possíveis entre zero (ausência de energia incidente) e 255 (saturação do sensor).

DV Variável discreta



$L_{min} \Rightarrow 0$

$L_{max} = 255$

$DN = a * L + b$

Ou outra função

L

Variável contínua

Para isto, é definida uma escala discreta de valores, com valores  $L_{min}$  e  $L_{max}$  fixos e uma quantidade de valores possíveis.

Esta quantidade é chamada de resolução radiométrica e, na prática, é definida pelo número de bits usados para armazenar a informação. Geralmente, são usados 8 bits para cada pixel, o que corresponde a um total de 256 diferentes níveis possíveis entre zero (ausência de energia incidente) e 255 (saturação do sensor).



8 bits (256 DN)

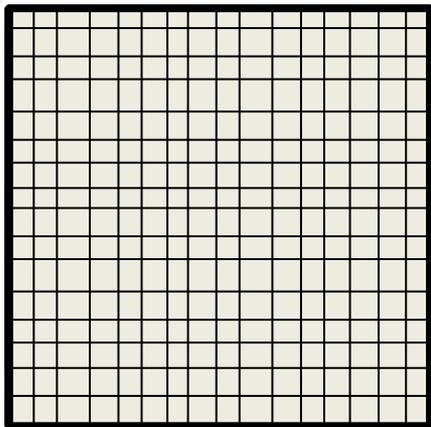


3 bits (8 DN)



1 bit (2 DN)

- Uma matriz = uma imagem nível de cinza



# Uma imagem simples

- Imagem de uma única banda.
- Os valores dos pixels são armazenados de maneira seqüencial. Esta organização requer apenas a preservação de uma ordem preestabelecida. Por exemplo, começando no pixel superior esquerdo e terminando no pixel inferior direito, efetuando a varredura da imagem de maneira seqüencial. estrutura espacial implícita, pois, conhecendo-se o tamanho da imagem formada pelos dados, é facilmente reconstruída a imagem.

1	2	3
4	5	6
7	8	9

Imagem tamanho 3x3

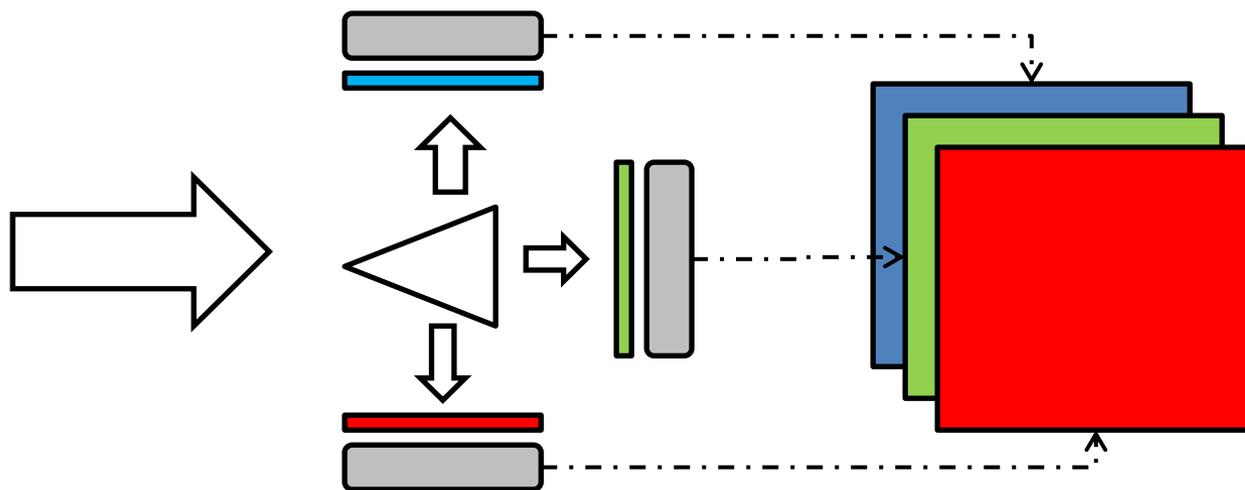
1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

Uma imagem colorida demanda o uso de três matrizes, uma para cada cor básica:  
RGB



# Bayer matrix

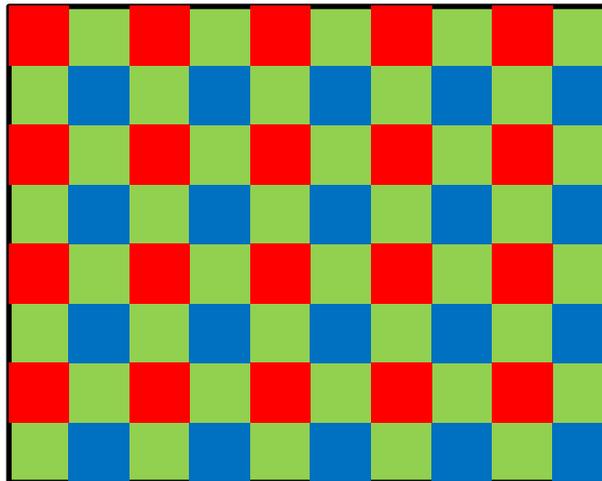
- Os sensores utilizados em câmaras (CCD /CMOS) são sensíveis à intensidade luminosa de todo o espectro (UV, Visível, parte do NIR), logo somente seria possível obter uma imagem em nível de cinza (pancromática)
- Para se obter uma imagem colorida seria necessário usar 3 sensores, um para cada cor, usando filtros de cores.



# Bayer matrix

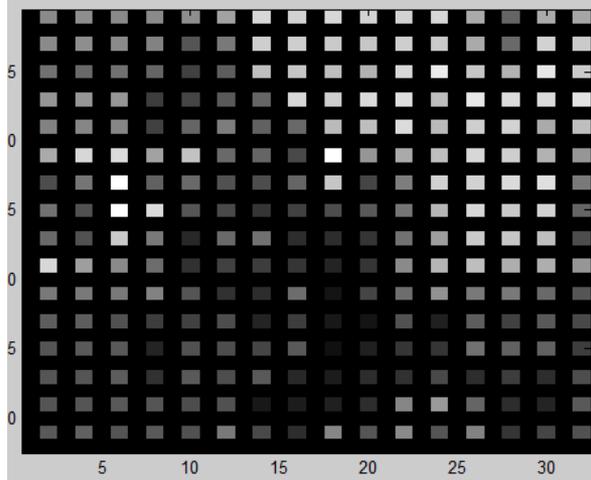
- Bryce E. Bayer, trabalhando para Kodak em 1974, propôs aplicar um filtro colorido a uma única matriz/sensor, seguindo um padrão xadrez contendo as cores RGB. Assim, cada elemento detector recebe a luz de uma cor específica e depois a imagem toda é interpolada.

Ken Parulski (Kodak): "*There are twice as many green elements as red or blue because this mimics the way the human eye provides the sharpest overall color image.*"

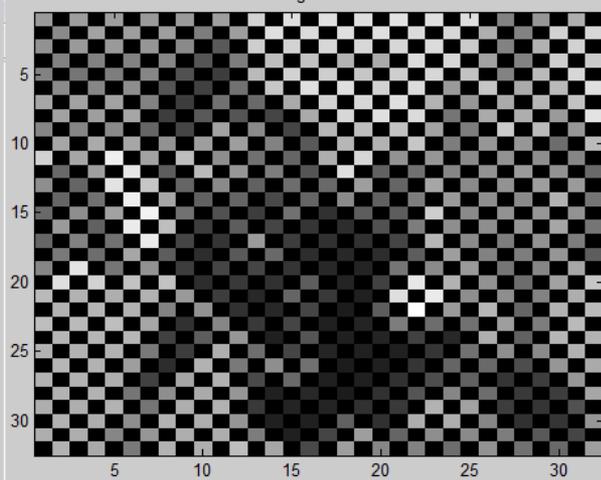




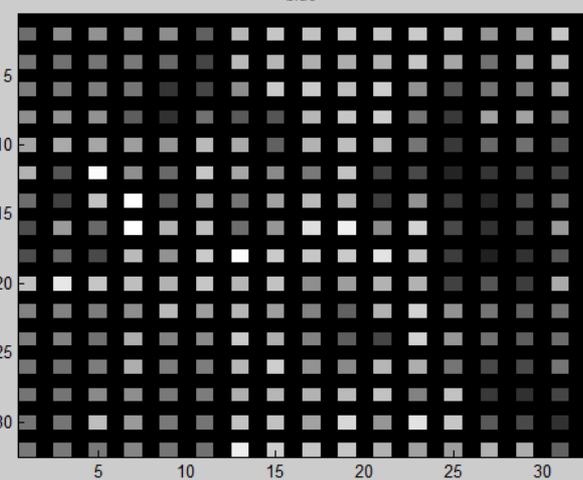
red



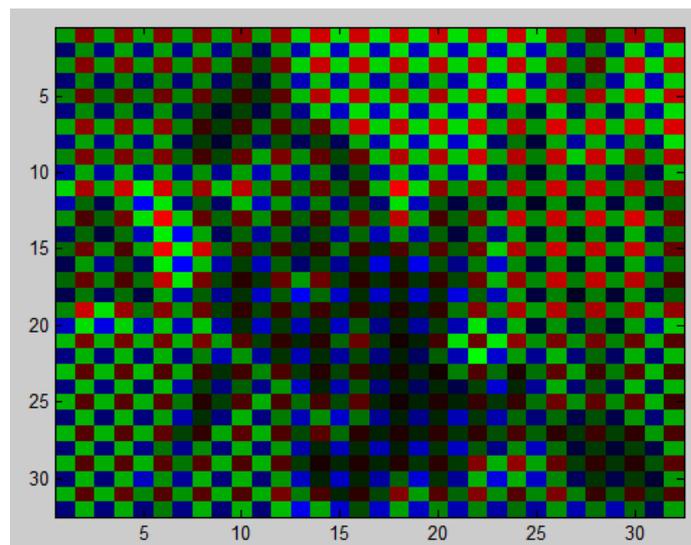
green



blue



- Interpolação de valores não preenchidos



# Uma imagem multispectral

- Quando se trata de imagens multispectrais, existem várias alternativas para o armazenamento dos valores de cada pixel nas diferentes bandas. As alternativas são:
- Intercalado por banda (BSQ), onde as bandas são armazenadas uma atrás da outra, sob forma de imagens de nível de cinza em um único arquivo.

1	2	3
4	5	6
7	8	9

1	2	3
4	5	6
7	8	9

1	2	3
4	5	6
7	8	9

1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

1	2	3
4	5	6
7	8	9

1	2	3
4	5	6
7	8	9

1	2	3
4	5	6
7	8	9

- Intercalado por linha (BIL): Os valores correspondentes a uma mesma linha nas diferentes bandas são armazenados juntos. Vantagem: pode ser lida uma linha inteira, em todas as bandas, de uma vez só.

1	2	3	1	2	3	1	2	3	4	5	6	4	5	6	4	5	6	7	8	9	7	8	9	7	8	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**Intercalado por pixel (BIP), onde todos os valores de um mesmo pixel, nas diferentes bandas, são armazenados juntos.**

1	1	1	2	2	2	3	3	3	4	4	4	5	5	5	6	6	6	7	7	7	8	8	8	9	9	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

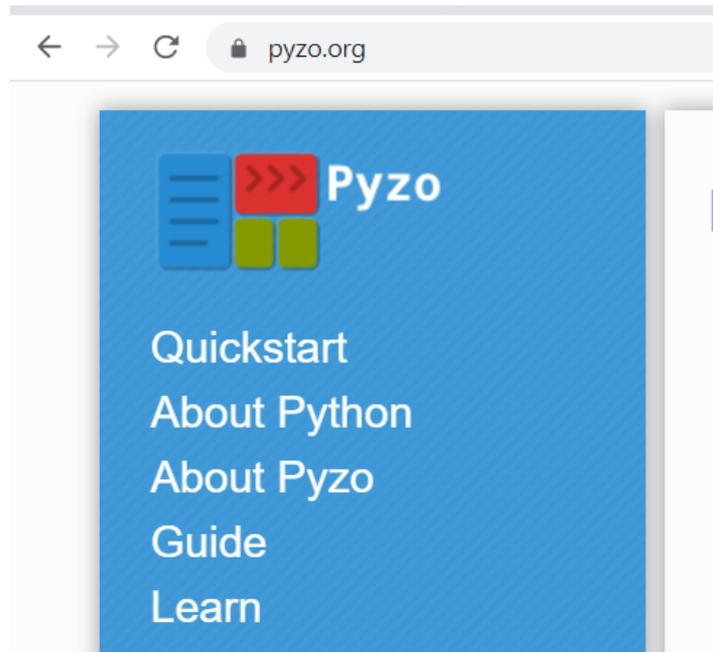
# Resolução

- A resolução de uma imagem é medida em termos de sua capacidade de representar a informação da cena.
- Resolução espacial (tamanho do pixel na superfície do objeto\)
- Resolução radiométrica (número de bits)
- Resolução espectral (bandas espectrais)



Aulas do professor Centeno - UFPR

# Breve introdução a Python com PYZO



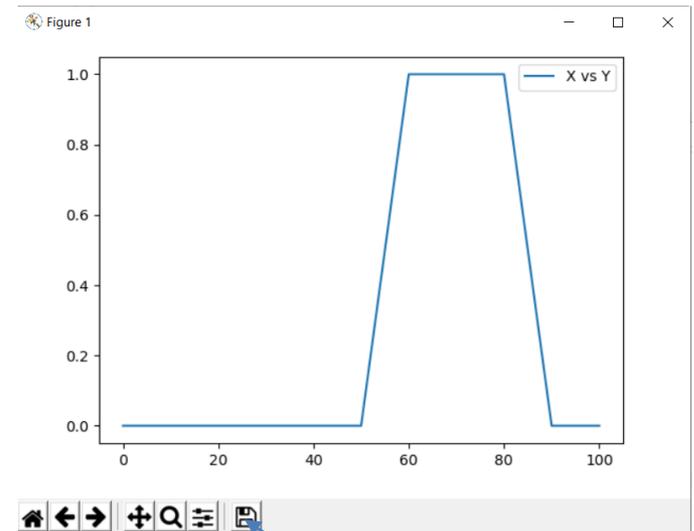


# Remarks

- Em Python, comentário começa com "cerquinha" #
- Ex:
- # inicio de programa
- Geralmente você inicia o Python básico, sem muitos recursos, mas como ele é modular, você pode e deve carregar as bibliotecas que for usar. Existem muitas. Isto é feito usando o comando **import**.
- Duas bibliotecas básicas são:
- **NumPy**: pacote que suporta arrays e matrizes multidimensionais, possui uma série de funções matemáticas para trabalhar com estas estruturas.
- **Matplotlib** pacote para apresentar gráficos e figuras na tela, junto com sua extensão **pyplot**.

# Um exemplo simples grafico x vs y

- `import numpy as np`
- `import matplotlib.pyplot as plt`
- `# definir dois vetores x e y`
- `x=[0, 50, 60, 80, 90, 100]`
- `y=[0, 0, 1, 1, 0, 0]`
- `# plotar o gráfico x vs y`
- `plt.plot(x,y)`
- `plt.show()`
- `# ou com legenda...`
- `plt.plot(x,y, label='X vs Y')`
- `plt.legend()`
- `plt.show()`

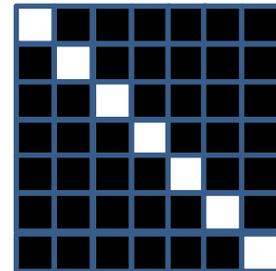
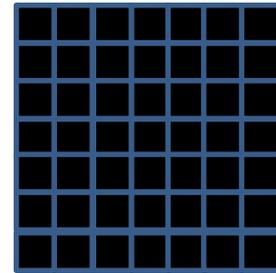


Aqui tem opções para “salvar” a imagem, dar zoom, etc

# Manipular imagens com PY

Vamos escrever um pequeno programa em PY para entender como se mudam os valores de uma imagem (uma matriz)

1. Criamos uma matriz vazia (ex. 7x7) com valores todos zero (preto)
2. Mudaremos os valores da diagonal principal para 255



Escreva o código com a ajuda a seguir e execute seu programa com `control+shift+E`

- # PROGRAMA basico em PY -criar uma imagem
- #####
- # primeiro: incluir algumas bibliotecas básicas
- **import** numpy as np
- **import** matplotlib.pyplot as plt
- # PY permite imprimir textos com "print"
- print('#####')
- print('# Programa inicial : criando uma imagem #')
- print('#####')
- # criamos matriz vazia (com valores reais FLOAT=zero, tamanho 17 x 17 pixels)
- I= np.zeros( (17,17) )
- # o comando shape permite ler as dimensões da matriz
- nl, nc = I.shape
- # imprimimos os valores número de linhas e colunas
- print("Linhas=", nl, " Colunas=", nc)
- # Mostrar o a matriz
- # a) montar a imagem como imshow [usando imshow= show image]
- # b) mostrar o grafico (SHOW)
- imshow = plt.imshow(I, cmap='gray', vmin=0, vmax=255)
- plt.show()

COMENTE as linhas que mostravam a imagem preta  
E continue o programa com isto...

- # Toda a imagem está cheia de zeros, vamos mudar os valores da diagonal para 255
- for L in range(n1):
- I[L,L]= 255
- # arredondamos e transformamos a matriz a uint 8 para armazenar como imagem
- P= np.uint8(np.round( I ))
- # salvar imagem como"saida.pgn"
- plt.imsave('saida.png',P,cmap='gray')
- # Mostrar o resultado
- # a) montar a imagem como imgplot [usando imshow= show image]
- # b) mostrar o grafico (SHOW)
- imgplot = plt.imshow(P, cmap='gray', vmin=0, vmax=255)

# RGB para PAN

- Agora, vamos ler uma imagem colorida com três bandas (R G B)
- Transformar a imagem para preto e branco (PAN)
- E salvar no disco



- # incluir algumas bibliotecas básicas
- import numpy as np
- import matplotlib.pyplot as plt
  
- print('#####')
- print('# Programa inicial : rgb to gray #')
- print('#####')
- # ler imagem usando uma função de matplotlib "PLT".imread
- I= plt.imread('rgb.tif')
- # SHAPE informa uma propriedade dos dados, sua dimensão
- # neste caso teremos nlinhas, ncolunas, e nbandas=3
- nl, nc, nb = I.shape
- print("Linhas=", nl, " Colunas-",nc, " Bandas=",nb)
  
- # podemos ver a imagem
- # a) montar a imagem como imgplot [usando imshow= show image]
- # b) mostrar o grafico (SHOW)
- imgplot = plt.imshow(I, cmap='gray', vmin=0, vmax=255)
- plt.show()

```
# Criamos uma matriz vazia (com valores reais FLOAT)
Y=np.array(I, dtype=float)
# agora varremos toda a imagem e somar os valores de r , g e b de cada pixel para calcular a
media

# CUIDADO com a indentação: define o domínio do loop
for L in range(nl):                                # varia as linhas
    for C in range (nc):                            # varia as colunas
        r=np.float( I[L,C,0] )
        g=np.float( I[L,C,1] )
        b=np.float( I[L,C,2] )
        media=(r+g+b)/3
        Y[L,C]= media # armazenamos o resultado na respectiva linha/coluna
# arredondamos os valores e transformamos a matriz a uint 8 para armazenar como imagem
P= np.uint8(np.round( Y ))
# salvar imagem como"saida.pgn"
plt.imsave('saida.png',P,cmap='gray')
# Mostrar o resultado
imgplot = plt.imshow(P, cmap='gray', vmin=0, vmax=255)
plt.show()
```

### Atividade:

- Obtenha uma imagem da parede com alvos no centro Politécnico com seu celular
- Transforme sua imagem a TIF
- Usando seu programa, transforme sua imagem a níveis de cinza (preto e branco)
- Responda:
  - Quantos pixels tem sua imagem original?
  - Quantos pixels tem sua imagem final?
  - Quantos bytes tem o arquivo original?
  - Quantos bytes tem o arquivo final?
  - Calcule a relação entre o número de pixels e a quantidade de bytes de cada imagem
  - Existe alguma explicação?

