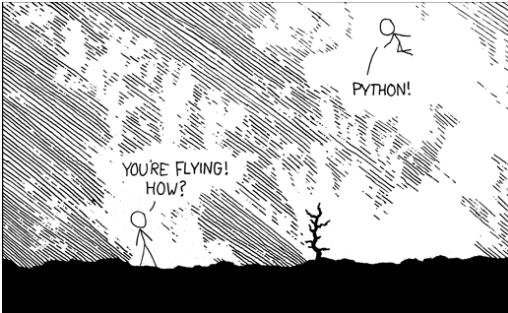




## Preliminares

- E-mail:  
– [hirata@ime.usp.br](mailto:hirata@ime.usp.br)
- Sala  
– B16
- Web:  
Material didático em:  
[www.adessowiki.org](http://www.adessowiki.org)
- [moodle](#) → [paca.ime.usp.br](http://paca.ime.usp.br)

## Python



## Python



## AdessoWiki

- [Curso não-presencial](#)
  - [FEE0146 – Laboratório de Programação Python/Numpy de Processamento de Imagens e Reconhecimento de Padrões](#)
  - [adessowiki.fee.unicamp.br/adesso-1/wiki/pirp\\_4e/view](http://adessowiki.fee.unicamp.br/adesso-1/wiki/pirp_4e/view)
- [www.adessowiki.org](http://www.adessowiki.org)
  - Algumas notas de aula
  - Exercícios

## Referências bibliográficas gerais

- Vision Science – S. Palmer – MIT Press
- Análise de Imagens Digitais – Pedrini e Schwartz – Editora Thomson
- Introdução ao Processamento de Imagens – Gonzalez e Woods

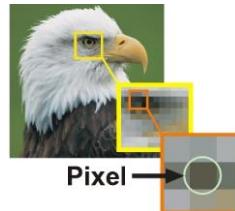
## Tentativa de Organização

- Aulas 1 e 2 (2 e 4 março) - Introdução, caracterização da imagem digital, brilho, contraste e análise de histograma, Python e AdessoWiki.
- Aulas 3 e 4 (9 e 11 de março) - Operações pontuais em imagens
- Aulas 5 e 6 (16 e 18 de março) - Topologia discreta e Morfologia Matemática
- Aulas 7 e 8 (23 e 25 de março) - Morfologia Matemática em níveis de cinza
- Aulas 9 e 10 (6 e 8 de abril) - Segmentação de imagens.
- Aulas 11 e 12 (13 e 15 de abril) - Operadores lineares
- Aulas 13 (22 de abril) - Extração de atributos locais e globais
- Aulas 14 e 15 (4 e 6 de maio) - Representação de Fourier
- Aulas 16 e 17 (11 e 13 de maio) - Princípios físicos da visão
- Aulas 18 e 19 (18 e 20 de maio) - Visão de cores
- Aulas 20 e 21 (25 e 27 de maio) - Teoria da informação
- Aulas 23 e 24 (8 e 10 de junho) - Textura e segunda prova
- Aulas 25 e 26 (15 e 17 de junho) - Visão de movimento

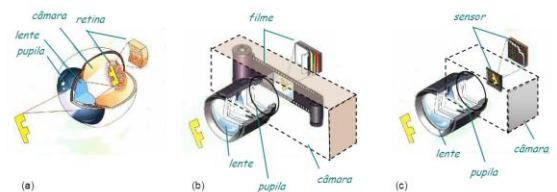
## Introdução

### Conceitos básicos sobre imagens

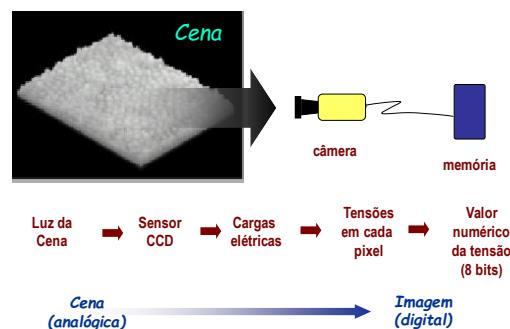
- Como a imagem se forma
- Propriedades da imagem
- Operações envolvendo a imagem



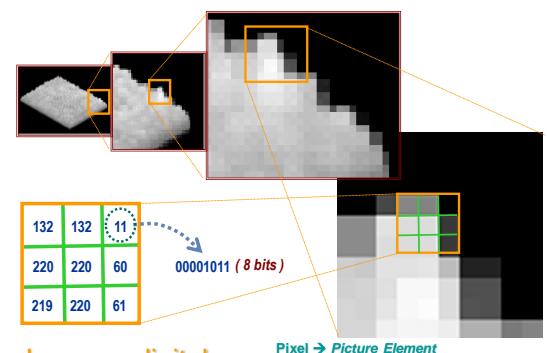
## Introdução



## Formação da imagem



### Formação das Imagens Digitais



### Imagen digital

## Intensidade – Imagens Monocromáticas

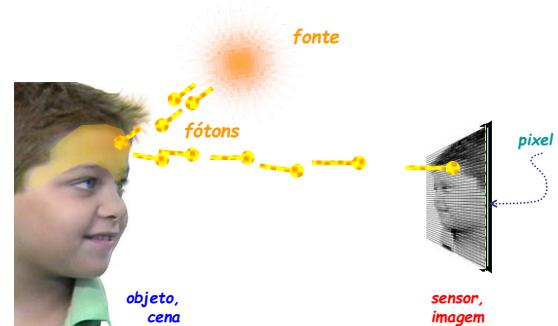
M := READBMH("img/Raphael-grey.bmp")



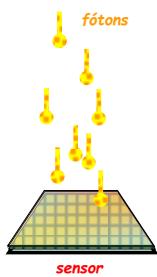
M

	0	1	2	3	4
0	240	240	241	240	241
1	240	240	241	241	242
2	240	241	240	240	240
3	239	241	240	240	239
4	239	240	241	240	240
5	240	241	241	241	240
6	241	241	241	241	242
7	243	242	241	241	242
8	242	242	242	241	241
9	243	242	242	241	243

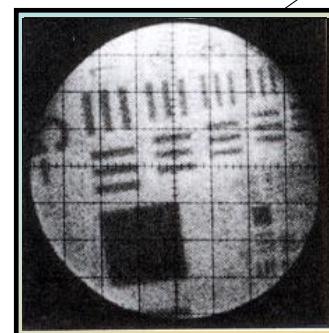
## Intensidade – Imagens Monocromáticas



## Intensidade – Imagens Monocromáticas

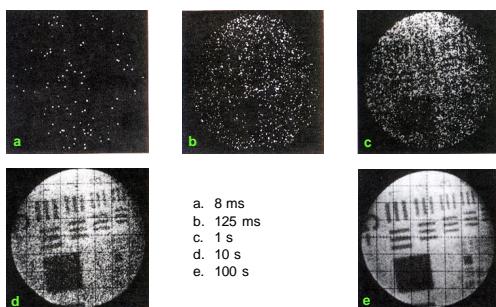


## Fótons



fotomultiplicadora  
fótons  
alvo  
Fonte de fótons (iluminação)

## Fótons



## Intensidade – Imagens Monocromáticas

Qual é o intervalo de valores numéricos para palavra de 16 bits ?

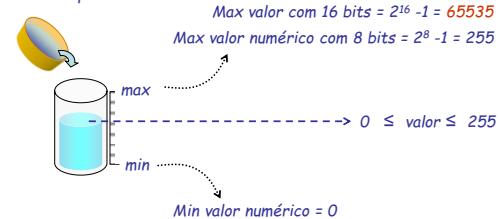
E de 32 bits ?

Intensidade em 1 pixel

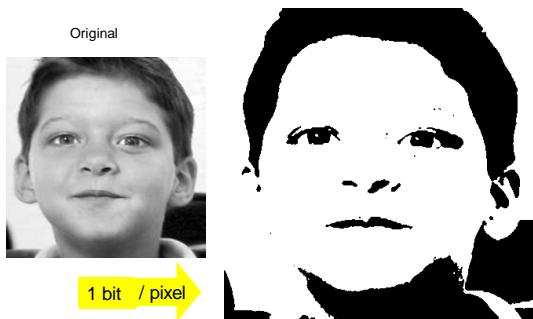
com 32 bits =  $2^{32} - 1 = 4,292,967,295$

Max valor com 16 bits =  $2^{16} - 1 = 65535$

Max valor numérico com 8 bits =  $2^8 - 1 = 255$



### Níveis de Intensidades



## Resolução em imagens digitais

### • Resolução em amplitude

- Resolução numérica do processo de conversão analógico → digital

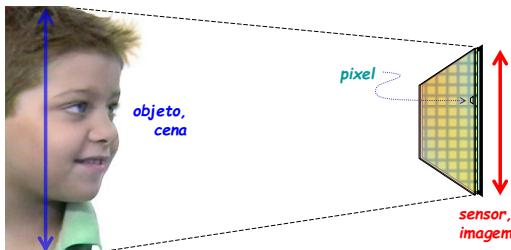
### • Resolução espacial

- Resolução do processo de amostragem espacial :

- Quantas amostras da cena são adquiridas por unidade de comprimento

## Resolução espacial

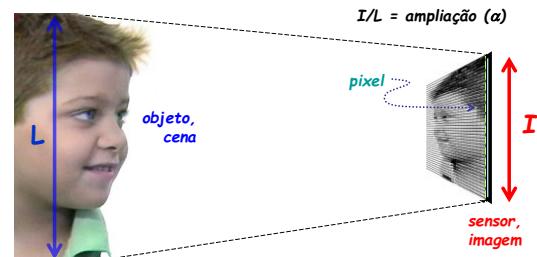
*Quantos pixels são necessários para cobrir uma cena de maneira eficiente?*



## Resolução espacial

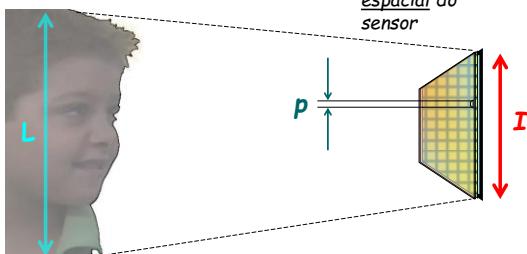
*Expressa em # pixels para cobrir a imagem = n*

*(o compromisso está em fazer L corresponder a I )*



## Resolução espacial

$$I = n \cdot p \quad \curvearrowleft n \rightarrow \# \text{pixels, ou resolução espacial do sensor}$$

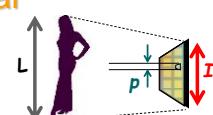


## Resolução espacial

### • Exercício

#### – Sabendo-se que:

- L = extensão maior da cena
- I = extensão maior do sensor
- p = extensão do pixel no sentido de I

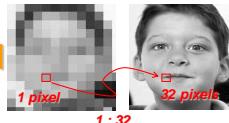


- Calcule o valor da *resolução física d* = tamanho do menor detalhe da cena visível na sua imagem formada no sensor, relacionando-a aos demais parâmetros acima.

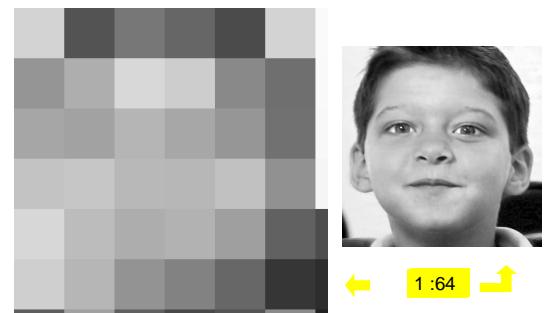
## Resolução espacial

- Exemplo

- É mostrada a imagem original
  - Que emprega toda a resolução disponível no sensor
- Em seguida são exibidas simulações de imagens obtidas com sensores de menor resolução espacial
  - Isto é, menos pixels disponíveis para cobrir a imagem
- Notação empregada:  $1:n$ 
  - Cada pixel da nova imagem compreende  $n \times n$  pixels da imagem original



Resolução Espacial



## Propriedades da imagem

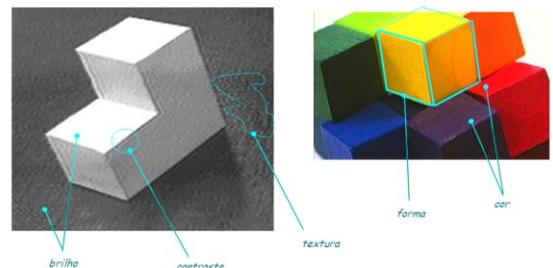
- Atributos físicos

- Resolução espacial
  - De quantos pixels é essa imagem ?
- Faixa dinâmica – ou resolução de amplitude
  - De quantos bits é essa imagem ?
- Tamanho
  - Quanta memória ocupa essa imagem ?
    - Tamanho = (Resolução) x (Faixa Dinâmica)

- Atributos perceptuais

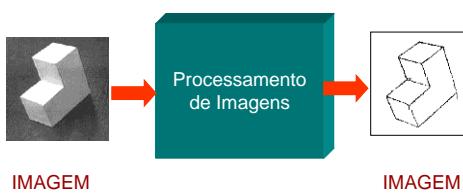
- Qualidades percebidas pela visão

## Atributos perceptuais da visão



## Processos

- Processamento de Imagens



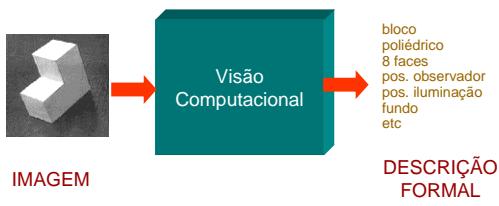
## Processos

- Computação Gráfica

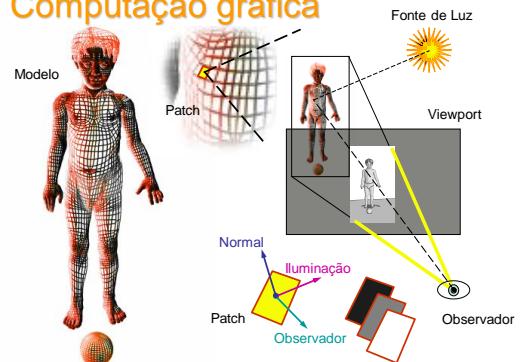


## Processos

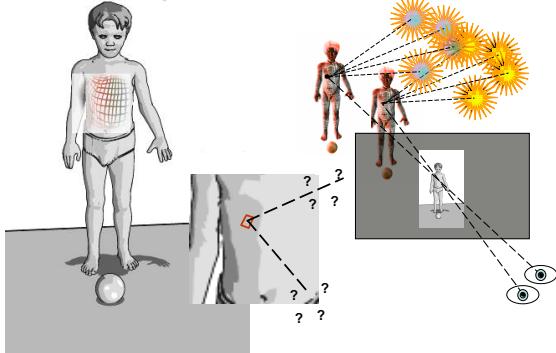
- Visão Computacional



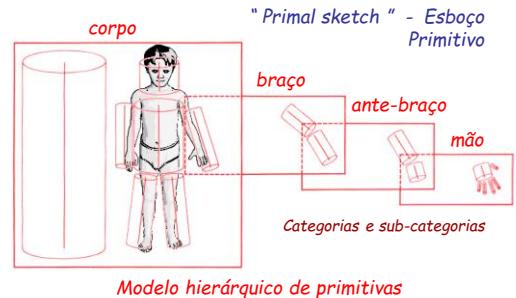
## Computação gráfica



## Visão Computacional → Problema Inverso

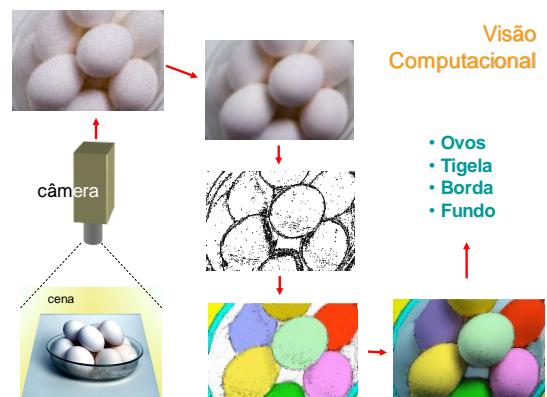


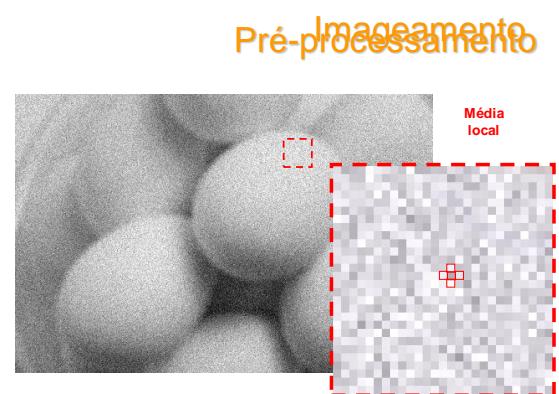
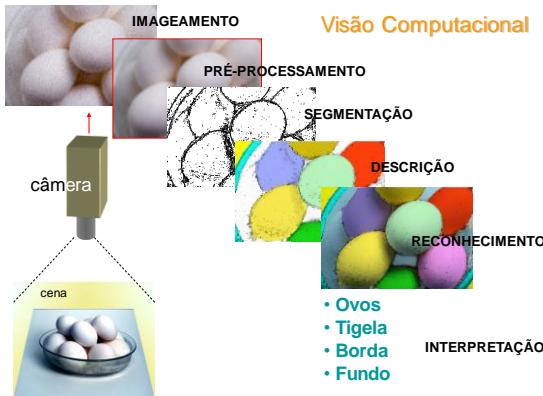
## Modelos e representação



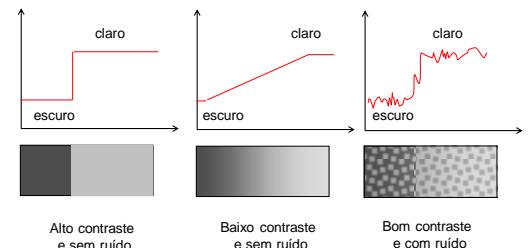
## Visão Computacional

- Lida com diversos tipos de problemas
  - Inspeção, identificação, monitoração, rastreamento, mensuração, etc
- Busca representações e modelos adequados a cada tipo de problema
  - Problemas de aplicação tecnológica
  - Problemas de modelagem comportamental e/ou fisiológica

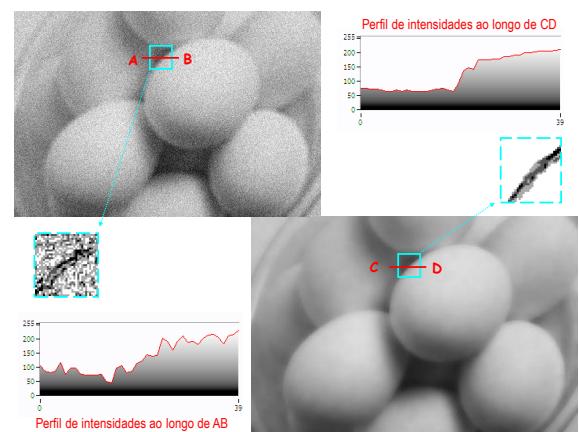
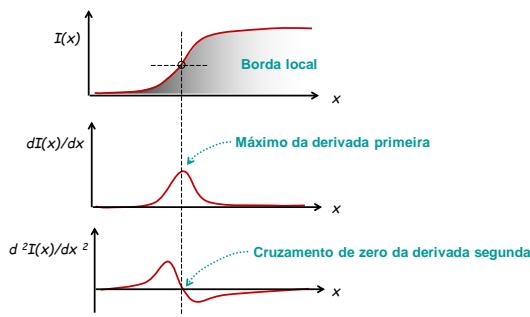


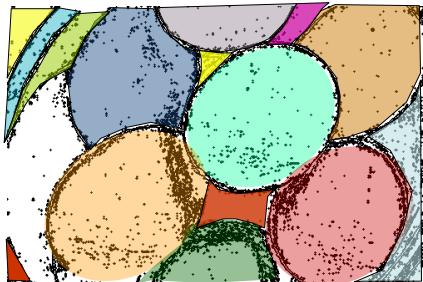


**Princípio da detecção de contorno pelo gradiente local**

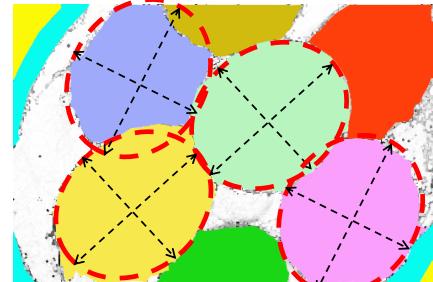


**Princípio da detecção de contorno pelo gradiente local**



Segmentação  
Descrição

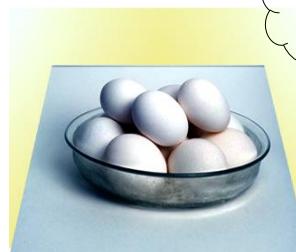
Descrição



Reconhecimento



Interpretação



conhecimento

### Decisões sobre informação visual

- Seleção de dados na cena
- Seleção de detalhes



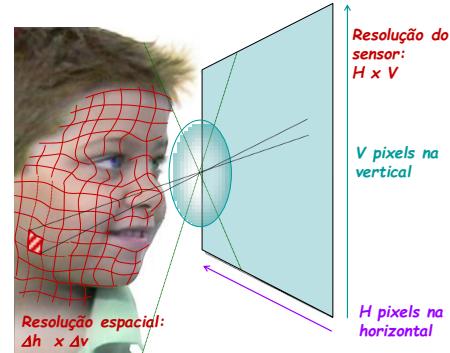
### Visão computacional

- Controlar a posição de uma câmera
  - Olhar para uma certa direção
  - Capturar dados de interesse
  - Selecionar as imagens a serem utilizadas
- Novos problemas
  - Como decidir que uma cena é mais interessante que outra ?
  - Quais dados na cena são importantes ?
  - Como inferir ou deduzir a partir de uma cena ?

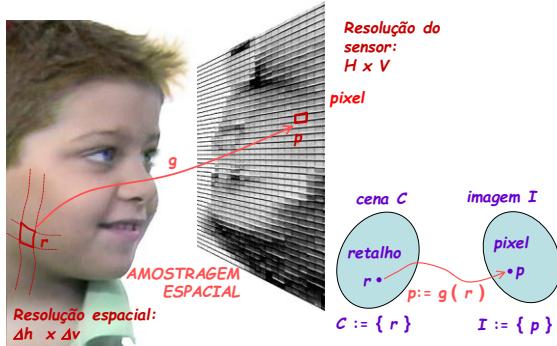


## Caracterização das imagens monocromáticas

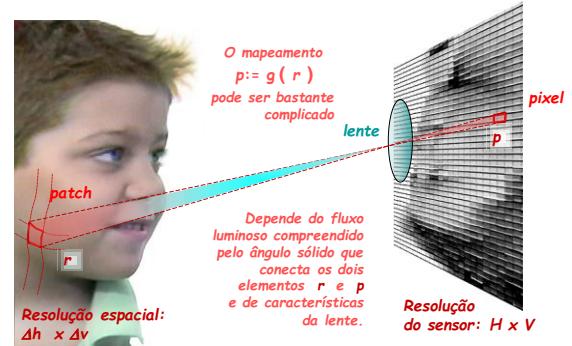
### Intensidade – Imagens Monocromáticas



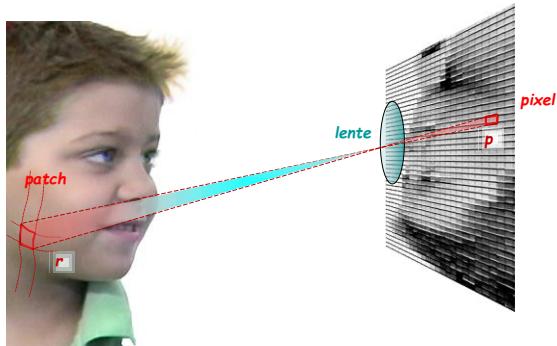
### Intensidade – Imagens Monocromáticas



### Intensidade – Imagens Monocromáticas



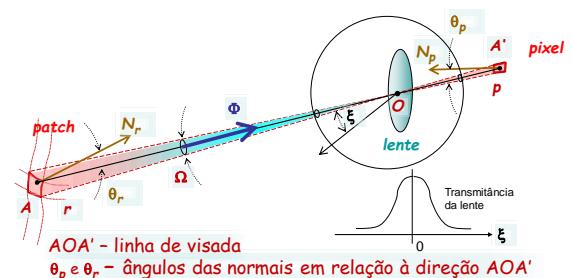
### Intensidade – Imagens Monocromáticas



$N_p$  e  $N_r$  – normais aos elementos  $p$  e  $r$

$\Omega$  – ângulo sólido compreendido por  $p$  e  $r$

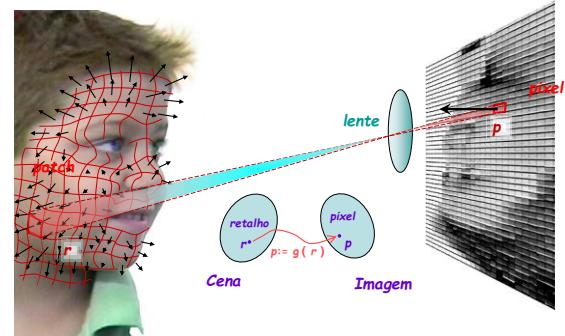
$\Phi$  – fluxo de energia radiante (vetor de Poynting)



## Intensidade – Imagens Monocromáticas

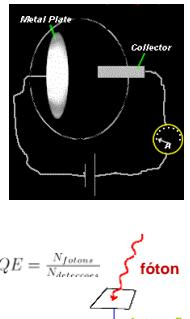
- Neste curso introdutório não procuraremos evidenciar como é a forma de  $p = g(r)$
- De fato esse mapeamento é uma distribuição, um funcional.
- Essa distribuição envolve a função de transmitância angular da lente aplicada sobre o fluxo de energia radiante do elemento  $r$  ao elemento  $p$  dentro do ângulo sólido que os conecta, mais o espalhamento sobre o plano do sensor (função de Green do espalhamento ondulatório do campo emergente da lente)

## Intensidade – Imagens Monocromáticas

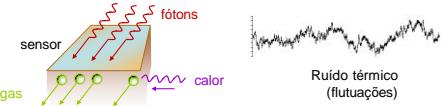


## Intensidade – Imagens Monocromáticas

- O resultado  $p = g(r)$  corresponde à energia que sai do elemento  $r$  e chega em  $p$  e vai produzir a carga armazenada no pixel, por efeito fotoelétrico.
- O sensor apresenta uma certa eficiência quântica, que é a capacidade de produzir cargas a partir de dado fluxo luminoso que chega.



## Intensidade – Imagens Monocromáticas



- Parte das cargas produzidas no sensor provêm de ruído térmico no mesmo
- Após isso a carga total no pixel será convertida em tensão elétrica, em função da capacidade do elemento sensor
- Essa tensão será convertida em valor numérico digital e armazenado na memória → esse é o valor da intensidade no pixel

## Radiometria e Fotometria

- Radiometria**
  - medidas de radiação emitida por uma fonte ou espalhadas por uma superfície
- Fotometria**
  - medidas de radiação emitida por uma fonte ou espalhadas por uma superfície percebidas pelo ser humano
  - A fotometria é perceptual
    - Os equipamentos fotométricos são calibrados de acordo com parâmetros perceptuais medidos de forma sistemática

## Unidades de medida

### Radiometria

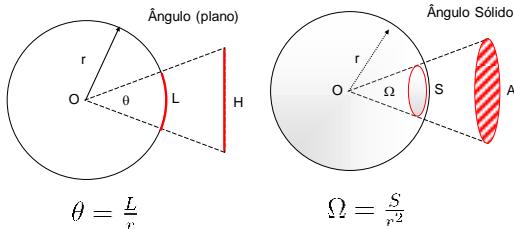
- Fluxo radiante → W (Watt)
- Irradiância → W / m<sup>2</sup>
- Intensidade radiante → W / sr (sr → sterad)
- Radiância → (W / sr) / m<sup>2</sup>

1 watt = 683 lumens

### Fotometria

- Fluxo luminoso → I (lúmen - lm)
- Iluminância → I / m<sup>2</sup> (lux - lx)
- Intensidade luminosa → I / sr (candela - cd)
- Luminância → I / sr / m<sup>2</sup> (cd / m<sup>2</sup>)

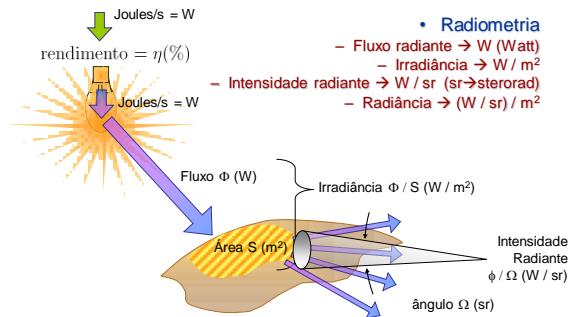
## Ângulo Sólido



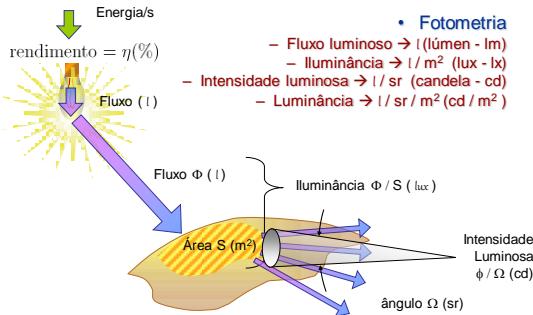
### Exercício

- Demonstre que o ângulo sólido que cobre a esfera toda vale  $4\pi$  (estero-radianos, sr).

## Unidades Radiométricas



## Unidades Fotométricas



## Fatos sobre intensidade

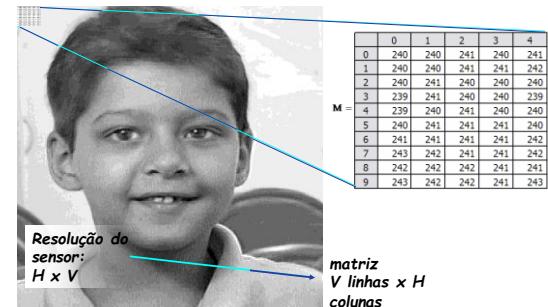
- Distribuição de luz natural:  $10000 \text{ cd/m}^2$
- Dist. Luz (amb. interno):  $100 \text{ cd/m}^2$
- Superfície clara (ambiente int.):  $50 \text{ cd/m}^2$
- Superfície escura (amb. interno):  $1 \text{ cd/m}^2$
- Faixa dinâmica requerida:  $1 \sim 10000 \text{ cd/m}^2$

Quantos bits por pixel seriam necessários para se discriminar valores nessa faixa dinâmica ?

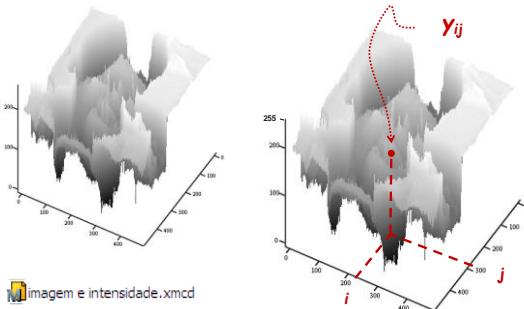
## Intensidade – Imagens Monocromáticas



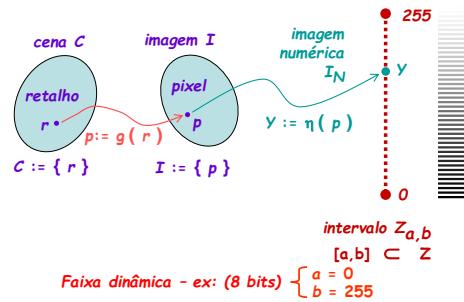
## Intensidade – Imagens Monocromáticas



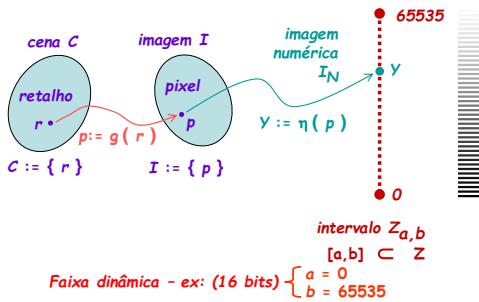
## Imagen digital



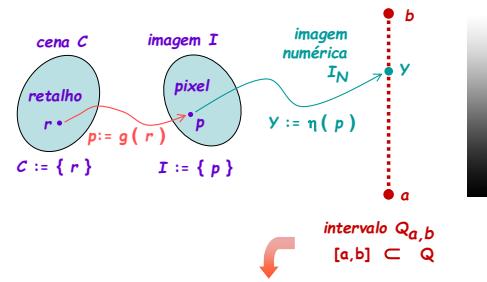
## Imagens Monocromáticas de 8 bits



## Imagens Monocromáticas de 16 bits



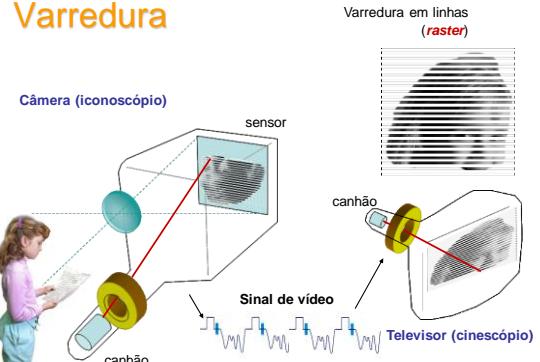
## Imagens Monocromáticas - Float



## Imagen → Sinal de Vídeo

- A tecnologia de imagem hoje empregada em computadores é **filha da tecnologia do vídeo em televisão**
- Em sua origem o vídeo em TV era produzido explorando-se as linhas da imagem formada em um sensor usando-se um feixe de elétrons
- A passagem do feixe de elétrons é chamada de **varredura**
- A informação resultante era um **sinal variável no tempo**, carregando o **conteúdo de luminosidade** em cada linha de varredura

## Varredura



## Varredura - 1

- Na câmera de TV (iconoscópio) a imagem era focalizada sobre um sensor
- O sensor acumulava cargas que eram lidas com um feixe
- Essa atividade chama-se exploração
- A exploração era feita linha a linha
- O processo de varrer a imagem linha a linha chama-se *rastering* (rasterização)

## Varredura - 2

- As linhas eram transmitidas ao receptor de TV via ondas curtas (MHz)
- O receptor exibe na tela do cinescópio também linha a linha, com um feixe de elétrons que excita material fotoluminescente aplicado sobre a tela (máscara de fósforo)
- O aparelho de TV e a câmera têm de estar sincronizados para que a imagem lida no emissor seja entendida no receptor
- Quando isso não acontecia ....

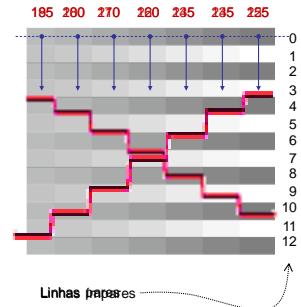
## Sinal de Vídeo (análogo)

A origem de diversos conceitos empregados hoje com imagens digitais provém da tecnologia analógica

- Muitos desses conceitos ainda são utilizados, bem como a própria tecnologia analógica (p.ex, em câmeras de segurança, emissoras de TV)
- As razões disso ser assim são:
  - Compatibilidade na troca de tecnologias
  - Mudança lenta dos conceitos
  - Preservação de conceitos julgados úteis
  - Custos “menores” da tecnologia analógica

## Sinal de Vídeo (análogo)

- Exemplo ilustrativo
  - Suponha que você tenha uma imagem composta da seguinte maneira: todas as linhas pares têm um degradê de cinzas em um sentido e as linhas ímpares têm o mesmo degradê só que em outro sentido



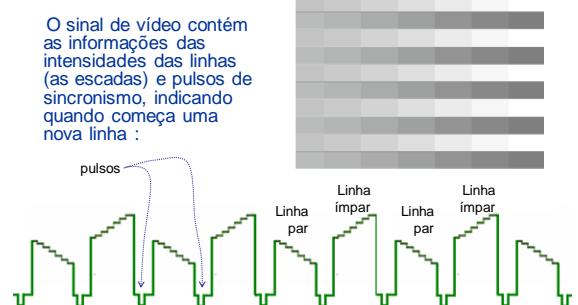
## Sinal de Vídeo (análogo)

- Isso poderia ser o sinal de vídeo, entretanto...
  - É preciso incluir a informação de sincronismo para que sua TV saiba como a câmera produziu a imagem



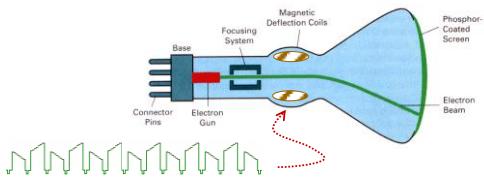
## Sinal de Vídeo (análogo)

O sinal de vídeo contém as informações das intensidades das linhas (as escadas) e pulsos de sincronismo, indicando quando começa uma nova linha :

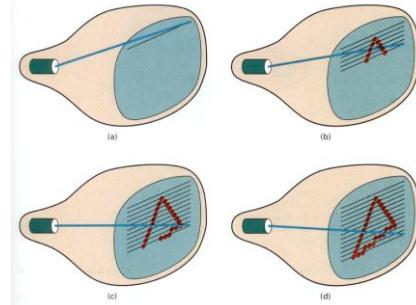


## Exibição do vídeo

- O vídeo é exibido lendo-se eletronicamente o sinal de vídeo e transformando-o no movimento do feixe de elétrons que varre a tela



## Varredura Rasterizada



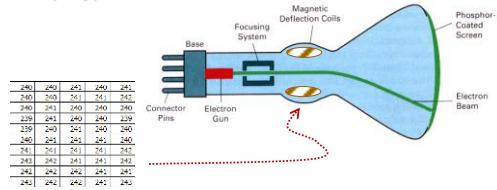
## Varredura - 3



- O sinal de vídeo **transporta** da câmera para o receptor a **informação de sincronismo**

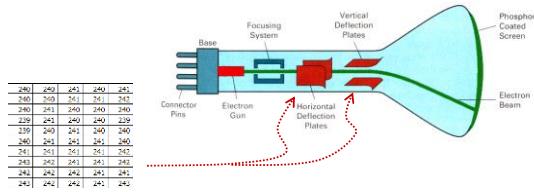
## Vídeo digital

- O vídeo é exibido lendo-se os dados da memória de vídeo e transformando-os no movimento do feixe de elétrons que varre a tela (**feito pelo controlador de vídeo + circuitos do monitor ou display**)

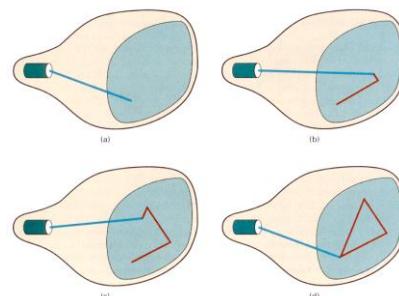


## Vídeo digital

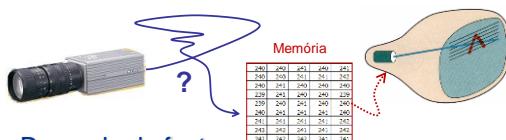
- Também existem monitores, muitas vezes de tecnologia usando tubo de raios catódicos (TRC) que operam de forma vetorial → varredura vetorializada



## Varredura Vetorizada (Vector Graphics)



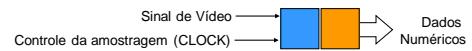
... mas, como os dados do vídeo digital chegam à memória ?



- Dependê da fonte
  - Câmera analógica x câmera digital
- Dependê da transmissão
  - Analógica – NTSC, PAL, outros
  - Digital – Ethernet, USB, Firewire, outros

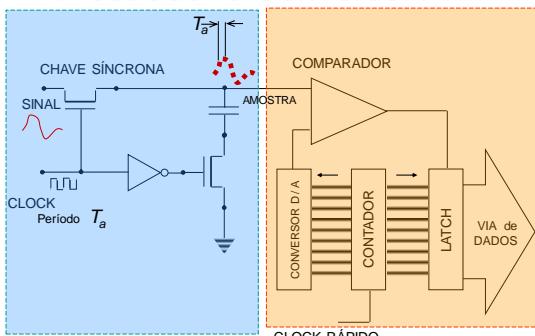
## Câmera Analógica

- O sinal de vídeo é analógico
  - É transportado em um cabo coaxial
- Precisa ser digitalizado
  - Placa de captura de vídeo analógico
    - Amostrador
    - Conversor Analógico/Digital



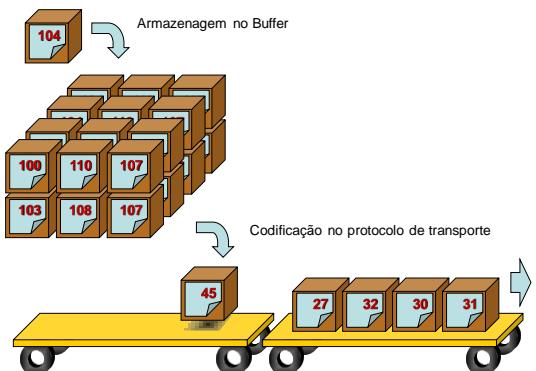
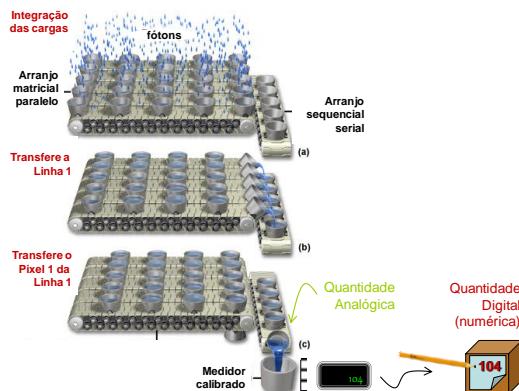
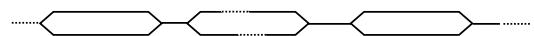
[Amostrador e Conversor A/D](#)

## Amostrador e Conversor A/D

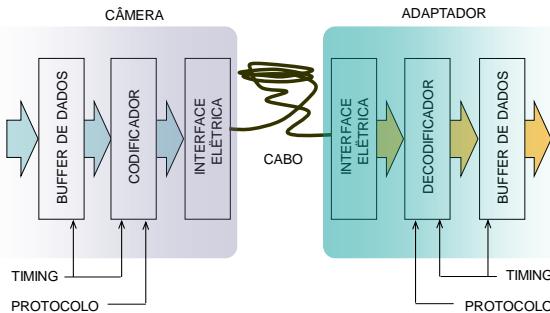


## Câmera Digital

- O sinal é digitalizado na câmera
- Os dados numéricos são transmitidos digitalmente segundo um protocolo dependente da interface usada
  - Ethernet → TCP
  - USB → Serial assíncrono
  - Firewire → Pacotes IEEE-1394



## Interface de Comunicação



## Interfaces de Vídeo Digital

Interface	Banda (MBytes/s)	Alcance Máximo (m)
Firewire - IEEE-1394a	50	4,5
USB 2.0	60	5 (sem hub)
Firewire-IEEE-1394b	100	4,5 (Cobre), 100 (fibra op.)
GigE	125	110
Camera Link	400	10
Barramento	Banda (MBytes/s)	-
PCI	128	-
PCI Express	4096	-

## Organização da imagem na memória

### 1. Estruturas de dados

- Vetores

- Matrizes

### 2. Varredura de uma imagem no sentido raster / anti-raster

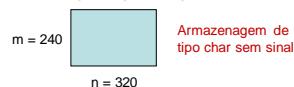
## Armazenagem na memória

- A armazenagem da imagem na memória é realizada através da **alocação** de um array
  - Unidimensional → vetor
  - Bidimensional → matriz
- Os arrays são alocados com o tipo de dado que corresponde ao tipo da imagem → inteiro de 8, 16, 32 ou 64 bits ou ainda ponto flutuante

## Armazenagem como Vetor

- A imagem pode ser armazenada como vetor (array unidimensional)

- Exemplo: (em C)



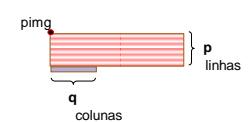
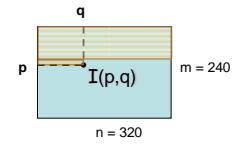
```
unsigned int m = 240, n = 320 ;
unsigned char *pimg ;
pimg = (unsigned char *) malloc(sizeof(char)*m*n) ;
if (pimg==NULL) { ERRO }
```

## Acesso como Vetor

- Acesso a um ponto  
 $P = I(p,q) , ( 0 \leq p , q \leq n )$   
 da imagem:

$$\text{value} = *(pimg + p*n + q)$$

para se chegar ao ponto P,  
 pulam-se p linhas e mais q colunas



Toda operação pontual sobre a imagem armazenada como vetor pode ser feita com um único loop

## Armazenagem como Matriz

- A imagem também pode ser armazenada como **matriz** (array bidimensional), que é a estrutura mais natural:

```
unsigned int i, m = 240, n = 320 ;
unsigned char **ppimg, *pimg ;

ppimg = (unsigned char **) malloc(sizeof(char**) * m) ;
if (ppimg == NULL) { Tratamento de erro }

for (i=0; i<m; i++) {
    ppimg[i] = (unsigned char *) malloc(sizeof(char*) * n) ;
    if (ppimg[i]==NULL) { Tratamento de erro mais complexo }
}
```

Diagrama de uma matriz 2D com  $m$  linhas e  $n$  colunas. As linhas são apontadas por  $m$  ponteiros, e cada linha tem  $n$  posições.

## Acesso como matriz

- Matriz:** uma operação pontual sobre um pixel  $I(p,q)$  da imagem pode ser feita com um simples acesso a `ppimg`,

$$\text{value} = \text{ppimg}[p][q]$$

Toda operação pontual sobre a imagem armazenada como matriz é feita com dois loops

## Arquivamento de imagens

- Há diversos formatos para se guardar as imagens em arquivos
- Os formatos conhecidos têm as mais diversas origens e razões
- Existem formatos para exploração raster e para exploração vetorizada

## Exemplo : Sun raster image header

```
• struct rasterfile {
    int ras_magic;
    int ras_width;
    int ras_height;
    int ras_depth;
    int ras_length;
    int ras_type;
    int ras_maptype;
    int ras_maplength;
}

#define RAS_MAGIC 0x59a66a95
```

## Sun raster

```
• /* Sun supported ras_type's */
• #define RT_OLD 0 /* Raw pixrect image in 68000 byte order */
• #define RT_STANDARD 1 /* Raw pixrect image in 68000 byte order */
• #define RT_BYTE_ENCODED 2 /* Run-length compress. of bytes */
• #define RT_FORMAT_RGB 3 /* XRGB or RGB instead of XBGR or BGR */
• #define RT_FORMAT_TIFF 4 /* tiff <> standard rasterfile */
• #define RT_FORMAT_IFF 5 /* iff (TAAC format) <> standard rasterfile */
• #define RT_EXPERIMENTAL 0xffff /* Reserved for testing */
• /* Sun registered ras_maptype's */

• #define RMT_RAW 2 /* Sun supported ras_maptype's */
• #define RMT_NONE 0 /* ras_maplength is expected to be 0 */
• #define RMT_EQUAL_RGB 1 /* red[ras_maplength/3].green[],blue[] */
```

## Sun raster

- ras\_length** field é o comprimento dos dados (comprimento do arquivo menos o comprimento do cabeçalho e do colormap)
  - Este valor é zero algumas vezes por questões de compatibilidade.
- ras\_maptype** e **ras\_maplength** contém o tipo e o comprimento do mapa de cores (colormap)
  - ras\_maptype** pode ser `RMT_NONE`, `RMT_RAW`, ou `RMT_EQUAL_RGB`
- Note que, independente de `width`, as linhas são armazenadas em múltiplos de 16bits.