

# Processamento de Imagens - PIM

Rogério Eduardo da Silva - [rogerio.silva@udesc.br](mailto:rogerio.silva@udesc.br)  
Gilmário Barbosa dos Santos - [gilmario.santos@udesc.br](mailto:gilmario.santos@udesc.br)

Universidade do Estado de Santa Catarina  
Departamento de Ciência da Computação

9 de outubro de 2017

# Ementa

- Fundamentos
- Operações globais e de vizinhança
- Transformadas
- Teorema da convolução
- Realce
- Restauração
- Segmentação
- Morfologia
- Reconhecimento
- Compressão
- Aplicações

# Objetivo Geral da Disciplina

- Apresentar os principais conceitos envolvidos na aquisição, processamento e análise de imagens digitais.

# Objetivo Geral da Disciplina

- Apresentar os principais conceitos envolvidos na aquisição, processamento e análise de imagens digitais.
- Introduzir os conceitos fundamentais das técnicas de processamento e codificação de imagem.

# Objetivo Geral da Disciplina

- Apresentar os principais conceitos envolvidos na aquisição, processamento e análise de imagens digitais.
- Introduzir os conceitos fundamentais das técnicas de processamento e codificação de imagem.
- Preparar os alunos para o uso de desenvolvimento de sistemas de processamento e análise de imagens.

# Objetivos Específicos

- Compreender os principais métodos de forma esquemática (em alto nível) e matemática para imagens digitais binárias e em níveis de cinza.

# Objetivos Específicos

- Compreender os principais métodos de forma esquemática (em alto nível) e matemática para imagens digitais binárias e em níveis de cinza.
- Implementar as técnicas mais simples e protótipos completos de aplicações em uma determinada área da ciência, selecionando informações importantes registradas em imagens digitais de forma semiautomática ou totalmente independente de interferência humana.

# Objetivos Específicos

- Compreender os principais métodos de forma esquemática (em alto nível) e matemática para imagens digitais binárias e em níveis de cinza.
- Implementar as técnicas mais simples e protótipos completos de aplicações em uma determinada área da ciência, selecionando informações importantes registradas em imagens digitais de forma semiautomática ou totalmente independente de interferência humana.
- Analisar diversos problemas de visão computacional e propor soluções aparentemente complexas em tempo mínimo, usando as ferramentas apresentadas.

# Objetivos Específicos

- Compreender os principais métodos de forma esquemática (em alto nível) e matemática para imagens digitais binárias e em níveis de cinza.
- Implementar as técnicas mais simples e protótipos completos de aplicações em uma determinada área da ciência, selecionando informações importantes registradas em imagens digitais de forma semiautomática ou totalmente independente de interferência humana.
- Analisar diversos problemas de visão computacional e propor soluções aparentemente complexas em tempo mínimo, usando as ferramentas apresentadas.
- Capacitar os alunos com embasamento teórico para trabalhos de pesquisa na área de processamento de imagens → LAPIS - Laboratório de Processamento de Imagens e Sinais.



<http://www2.joinville.udesc.br/~lapis>

# Método de Ensino

- Aulas expositivas em sala e em laboratório
- Listas de exercícios teóricos e práticos
- Atendimento presencial (sala do professor) e/ou através da lista de emails da disciplina

bcc-pim@googlegroups.com

- <http://www.rogerioesilva.net/>
- <http://www2.joinville.udesc.br/~lapis>

# Avaliações

- Participação em Classe;
- Provas (2 provas previstas – individuais e s/ consulta);
- Trabalhos individuais ou em grupos de 2 ou mais alunos, com o desenvolvimento de soluções para problemas sugeridos;

$$\text{NotaFinal} = \text{Pr}_1 * 0.4 + \text{Pr}_2 * 0.4 + \text{TF} * 0.2$$

*Pr<sub>1</sub>* Prova 1

*Pr<sub>2</sub>* Prova 2

*TF* Trabalho Final da Disciplina

# Conteúdo Programático I

## 1 Apresentação

## 2 Introdução

- Análise de Imagens
- Áreas de Aplicação
- Etapas do Processamento de Imagens

## 3 Fundamentos de Imagens Digitais

- Tipos de Sinais
- Modelo de Imagens
- Digitalização
- Resolução Espacial vs Profundidade de Imagem
- Representação de Imagens Digitais
- Imagens Multibandas ou Multiespectrais
- Imagens Multidimensionais
- Sistema Visual Humano
- Ruídos em Imagens
- Relacionamento entre Elementos de uma Imagem

## 4 Realce

## 5 Segmentação

## 6 Morfologia Matemática

# Bibliografia Básica Sugerida I

-  H. Pedrini and W. R. Schwartz, *Análise de Imagens Digitais – Princípios, Algoritmos e Aplicações*.  
Editora Thomson, 2008.
-  R. C. Gonzalez and R. E. Woods, *Processamento de Imagens Digitais*.  
São Paulo: Edgard Blücher, 2000.
-  L. O'Gorman, M. J. Sammon, and M. Seul, *Algorithms for Image Analysis: Description, Examples, Programs, and Projects*.  
New York: Cambridge University Press, 2<sup>nd</sup> ed., 2009.
-  G. A. Baxes, *Digital Image Processing: Principles and Applications*.  
John Wiley & Sons, 1<sup>st</sup> ed., 1994.
-  A. C. Bovik, ed., *Handbook of Image and Video Processing*.  
Academic Press, 1<sup>st</sup> ed., 2003.

# Bibliografia Básica Sugerida II

-  E. R. Dougherty and R. A. Lotufo, *Hands-on Morphological Image Processing*.  
 SPIE Press, 2003.
-  J. Gomes and L. Velho, *Computação Gráfica: Imagem*.  
 IMPA/SBM, 2<sup>nd</sup> ed., 2002.
-  J. Kiusalass, *Numerical Methods in Engineering with Python*.  
 Cambridge University Press, 2005.
-  B. Jähne, *Digital Image Processing*.  
 Springer, 5<sup>th</sup> ed., 2005.
-  A. K. Jain, *Fundamentals of Digital Image Processing*.  
 Englewood Cliffs, US: Prentice-Hall, 1988.
-  J. R. Parker, *Algorithms for Image Processing and Computer Vision – Bk&CD-Rom*.  
 John Wiley & Sons, 1996.

# Bibliografia Básica Sugerida III

-  W. K. Pratt, *Digital Image Processing: PIKS Inside.*  
John Wiley & Sons, 3<sup>rd</sup> ed., 2001.
-  J. C. Russ, *The Image Processing Handbook.*  
CRC, 5<sup>th</sup> ed., 2006.
-  G. J. F. Banon and J. Barrera, "Bases da morfologia matemática para análise de imagens."  
<http://www.vision.ime.usp.br/~jb/books/conteudo.pdf>,  
1998.  
Acessado em 2017-07-12.
-  A. Falcão, "Fundamentos de processamento de imagem digital."  
<http://www.dcc.unicamp.br/~cpg/materialdidatico/mo815/9802/curso/curso.html>.  
Acessado em 2017-07-12.

# Bibliografia Básica Sugerida IV

-  R. B. Fischer, "CV online: The evolving, distributed, non-proprietary, on-line compendium of computer vision."  
<http://homepages.inf.ed.ac.uk/rbf/CVonline/>.  
Acessado em 2017-07-12.
-  K. Price, "Annotated computer vision bibliography: Table of contents." <http://iris.usc.edu/Vision-Notes/bibliography/contents.html>, 2006.  
Acessado em 2017-07-12.

# Introdução

# Introdução

## Desafio

Desenvolvimento de sistemas autônomos que reproduzam as capacidades do sistema visual humano e que sejam capazes de reagir a estímulos visuais de forma adequada à área específica sob investigação.

# Introdução

## Desafio

Desenvolvimento de sistemas autônomos que reproduzam as capacidades do sistema visual humano e que sejam capazes de reagir a estímulos visuais de forma adequada à área específica sob investigação.

## Visão Computacional

Procura auxiliar a resolução de problemas altamente complexos, buscando imitar a cognição humana e a habilidade do ser humano em tomar decisões de acordo com as informações contidas na imagem

# Introdução

## Desafio

Desenvolvimento de sistemas autônomos que reproduzam as capacidades do sistema visual humano e que sejam capazes de reagir a estímulos visuais de forma adequada à área específica sob investigação.

## Visão Computacional

Procura auxiliar a resolução de problemas altamente complexos, buscando imitar a cognição humana e a habilidade do ser humano em tomar decisões de acordo com as informações contidas na imagem

## Processamento Digital de Imagens

Consiste em um conjunto de técnicas para capturar, representar e transformar imagens com o auxílio do computador

# Introdução

## Análise de Imagens

- O emprego dessas técnicas permite extrair e identificar informações das imagens e melhorar a qualidade visual de certos aspectos estruturais, facilitando a percepção humana e a interpretação automática por meio de máquinas

# Introdução

## Análise de Imagens

- O emprego dessas técnicas permite extrair e identificar informações das imagens e melhorar a qualidade visual de certos aspectos estruturais, facilitando a percepção humana e a interpretação automática por meio de máquinas
- A **análise** ou a **interpretação** de imagens visa obter uma descrição que contenha informação suficiente para distinguir entre diferentes objetos de interesse, de forma confiável e requerendo o mínimo de intervenção humana

# Introdução

## Análise de Imagens

- O emprego dessas técnicas permite extrair e identificar informações das imagens e melhorar a qualidade visual de certos aspectos estruturais, facilitando a percepção humana e a interpretação automática por meio de máquinas
- A **análise** ou a **interpretação** de imagens visa obter uma descrição que contenha informação suficiente para distinguir entre diferentes objetos de interesse, de forma confiável e requerendo o mínimo de intervenção humana
- A análise de imagens é, tipicamente, baseada na forma, na textura, nos níveis de cinza ou nas cores dos objetos presentes nas imagens

# Introdução

## Áreas de Aplicação

- Medicina
- Microscopia
- Biologia
- Automação Industrial
- Sensoriamento Remoto
- Astronomia
- Área Militar
- Segurança e Vigilância
- Computação Forense
- Arqueologia e Artes

# Introdução

Áreas de Aplicação: Medicina

- Diagnósticos médicos por imagem: raio-X, tomografia computadorizada, ressonância magnética, ultrassonografia

# Introdução

Áreas de Aplicação: Medicina

- Diagnósticos médicos por imagem: raio-X, tomografia computadorizada, ressonância magnética, ultrassonografia
- Diversas áreas médicas de interesse: oncologia, ortopedia, cardiologia, etc.

# Introdução

Áreas de Aplicação: Medicina

- Diagnósticos médicos por imagem: raio-X, tomografia computadorizada, ressonância magnética, ultrassonografia
- Diversas áreas médicas de interesse: oncologia, ortopedia, cardiologia, etc.
- Permitem a identificação de lesões, câncer, partes do corpo; de maneira a facilitar diagnósticos e planejamento de cirurgias e tratamentos.

# Introdução

Áreas de Aplicação: Microscopia

- Análise de imagens obtidas pela captura através de microscópios óticas ou eletrônicos

# Introdução

Áreas de Aplicação: Microscopia

- Análise de imagens obtidas pela captura através de microscópios óticas ou eletrônicos
- Áreas como biologia ou metalurgia

# Introdução

Áreas de Aplicação: Microscopia

- Análise de imagens obtidas pela captura através de microscópios óticas ou eletrônicos
- Áreas como biologia ou metalurgia
- Exemplos:
  - Contagem e identificação de células sanguíneas

# Introdução

Áreas de Aplicação: Microscopia

- Análise de imagens obtidas pela captura através de microscópios óticas ou eletrônicos
- Áreas como biologia ou metalurgia
- Exemplos:
  - Contagem e identificação de células sanguíneas
  - Análise de estruturas de cristalografia

# Introdução

Áreas de Aplicação: Microscopia

- Análise de imagens obtidas pela captura através de microscópios óticas ou eletrônicos
- Áreas como biologia ou metalurgia
- Exemplos:
  - Contagem e identificação de células sanguíneas
  - Análise de estruturas de cristalografia
  - Sequenciamento e análise de genes

# Introdução

Áreas de Aplicação: Automação Industrial

- Montagem e inspeção de produtos, visão robótica, controle de qualidade

# Introdução

Áreas de Aplicação: Automação Industrial

- Montagem e inspeção de produtos, visão robótica, controle de qualidade
- Atividades comuns:
  - Verificação de falhas

# Introdução

Áreas de Aplicação: Automação Industrial

- Montagem e inspeção de produtos, visão robótica, controle de qualidade
- Atividades comuns:
  - Verificação de falhas
  - Separação de peças em uma linha de montagem

# Introdução

Áreas de Aplicação: Automação Industrial

- Montagem e inspeção de produtos, visão robótica, controle de qualidade
- Atividades comuns:
  - Verificação de falhas
  - Separação de peças em uma linha de montagem
  - Classificação de defeitos em soldas

# Introdução

Áreas de Aplicação: Sensoriamento Remoto

- Análise de fotos aéreas ou imagens de satélite:

# Introdução

Áreas de Aplicação: Sensoriamento Remoto

- Análise de fotos aéreas ou imagens de satélite:
  - Áreas urbanas

# Introdução

Áreas de Aplicação: Sensoriamento Remoto

- Análise de fotos aéreas ou imagens de satélite:
  - Áreas urbanas
  - Análise de áreas de desmatamento

# Introdução

Áreas de Aplicação: Sensoriamento Remoto

- Análise de fotos aéreas ou imagens de satélite:
  - Áreas urbanas
  - Análise de áreas de desmatamento
  - Previsão de safras

# Introdução

Áreas de Aplicação: Sensoriamento Remoto

- Análise de fotos aéreas ou imagens de satélite:
  - Áreas urbanas
  - Análise de áreas de desmatamento
  - Previsão de safras
  - Extração de feições cartográficas: estradas, rios, edificações, etc.

# Introdução

Áreas de Aplicação: Área Militar

- Identificação e rastreio de potenciais alvos

# Introdução

Áreas de Aplicação: Área Militar

- Identificação e rastreio de potenciais alvos
- Navegação de veículos autônomos

# Introdução

Áreas de Aplicação: Área Militar

- Identificação e rastreio de potenciais alvos
- Navegação de veículos autônomos
- Detecção de obstáculos no caminho de robôs e drones

# Introdução

Áreas de Aplicação: Segurança e Vigilância

- Identificação biométrica: impressões digitais, faces, silhuetas, etc.

# Introdução

Áreas de Aplicação: Segurança e Vigilância

- Identificação biométrica: impressões digitais, faces, silhuetas, etc.
- Reconhecimento de assinaturas e caligrafia

# Introdução

Áreas de Aplicação: Segurança e Vigilância

- Identificação biométrica: impressões digitais, faces, silhuetas, etc.
- Reconhecimento de assinaturas e caligrafia
- Reconhecimento automático de placas de veículos

# Introdução

Áreas de Aplicação: Computação Forense

- Métodos científicos para preservação, coleta, validação, identificação, análise, interpretação, documentação e apresentação de evidências derivadas de meios digitais com validade probatória em juízo para facilitar a reconstrução de eventos, normalmente de natureza criminal

# Introdução

Áreas de Aplicação: Computação Forense

- Métodos científicos para preservação, coleta, validação, identificação, análise, interpretação, documentação e apresentação de evidências derivadas de meios digitais com validade probatória em juízo para facilitar a reconstrução de eventos, normalmente de natureza criminal
- Análise de dados: documentos, imagens, vídeos

# Introdução

Áreas de Aplicação: Arqueologia e Artes

- Restauração de artefatos antigos

# Introdução

Áreas de Aplicação: Arqueologia e Artes

- Restauração de artefatos antigos
- Pinturas e documentos antigos

# Introdução

Áreas de Aplicação: Arqueologia e Artes

- Restauração de artefatos antigos
- Pinturas e documentos antigos
- Museus virtuais

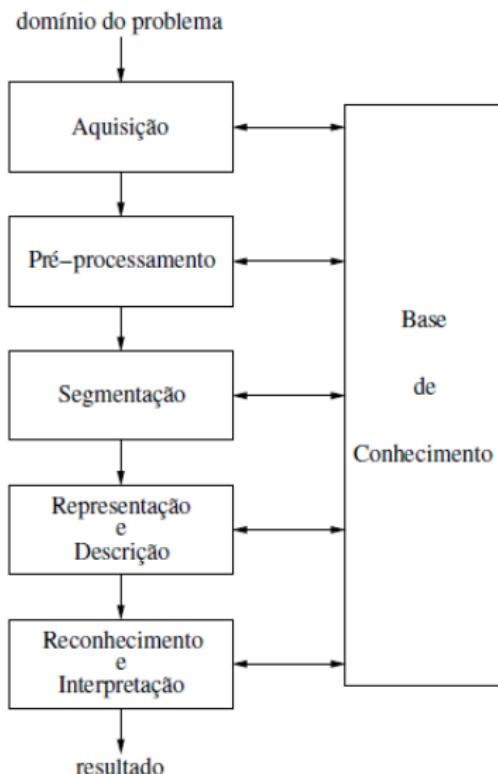
# Introdução

Áreas de Aplicação: Arqueologia e Artes

- Restauração de artefatos antigos
- Pinturas e documentos antigos
- Museus virtuais
- Manipulação de fotografia digital

# Introdução

## Etapas do Processamento de Imagens



# Introdução

## Domínio do Problema e Base de Conhecimento

- Um sistema de processamento digital de imagens é constituído de etapas, capazes de produzir um resultado a partir do **domínio do problema**

# Introdução

## Domínio do Problema e Base de Conhecimento

- Um sistema de processamento digital de imagens é constituído de etapas, capazes de produzir um resultado a partir do **domínio do problema**
- O conhecimento sobre o domínio do problema está codicado em um sistema de processamento de imagens na forma de uma **base de conhecimento**.

# Introdução

## Aquisição

- Etapa responsável pela captura de imagens por meio de dispositivos ou sensores e posterior conversão para uma representação adequada para posterior processamento

# Introdução

## Aquisição

- Etapa responsável pela captura de imagens por meio de dispositivos ou sensores e posterior conversão para uma representação adequada para posterior processamento
- Dispositivos de captura: câmeras de vídeo, tomógrafos, satélites, scanners, etc.

# Introdução

## Aquisição

- Etapa responsável pela captura de imagens por meio de dispositivos ou sensores e posterior conversão para uma representação adequada para posterior processamento
- Dispositivos de captura: câmeras de vídeo, tomógrafos, satélites, scanners, etc.
- Dispositivos sensíveis a uma certa banda do espectro eletromagnético, tais como raios-X ou raios infravermelhos, produzem um sinal elétrico de saída proporcional ao nível de energia detectado. Esse sinal elétrico é convertido em informação digital, tornando possível sua interpretação por meio de computadores

# Introdução

## Aquisição

- Etapa responsável pela captura de imagens por meio de dispositivos ou sensores e posterior conversão para uma representação adequada para posterior processamento
- Dispositivos de captura: câmeras de vídeo, tomógrafos, satélites, scanners, etc.
- Dispositivos sensíveis a uma certa banda do espectro eletromagnético, tais como raios-X ou raios infravermelhos, produzem um sinal elétrico de saída proporcional ao nível de energia detectado. Esse sinal elétrico é convertido em informação digital, tornando possível sua interpretação por meio de computadores
- Aspectos a serem considerados:
  - Tipo de Sensor
  - Condições de iluminação
  - Resolução gráfica
  - Quantidade de níveis de cinza ou cores

# Introdução

## Pré-Processamento

- A etapa de aquisição pode produzir imperfeições ou degradações decorrentes, por exemplo, das condições de iluminação ou características do dispositivo

# Introdução

## Pré-Processamento

- A etapa de aquisição pode produzir imperfeições ou degradações decorrentes, por exemplo, das condições de iluminação ou características do dispositivo
- Esta etapa visa melhorar a qualidade da imagem por meio de:
  - Técnicas de atenuação de ruído
  - Correção de contraste ou brilho
  - Suavização de características indesejáveis

# Introdução

## Segmentação

- Realiza a extração e identificação de áreas de interesse contidas na imagem

# Introdução

## Segmentação

- Realiza a extração e identificação de áreas de interesse contidas na imagem
- Baseia-se na detecção de descontinuidades (bordas) ou de similaridades (regiões) na imagem

# Introdução

## Representação e Descrição

- Estruturas adequadas de **representação** devem ser utilizadas para armazenar e manipular os objetos de interesse extraídos da imagem

# Introdução

## Representação e Descrição

- Estruturas adequadas de **representação** devem ser utilizadas para armazenar e manipular os objetos de interesse extraídos da imagem
- Imagens e vídeos requerem alta capacidade de armazenamento:
  - Uma imagem colorida  $1024 \times 1024$  *pixels* onde cada pixel seja representado por 24 bits, requer 3 Mb para seu armazenamento (sem compressão)

# Introdução

## Representação e Descrição

- Estruturas adequadas de **representação** devem ser utilizadas para armazenar e manipular os objetos de interesse extraídos da imagem
- Imagens e vídeos requerem alta capacidade de armazenamento:
  - Uma imagem colorida  $1024 \times 1024$  pixels onde cada pixel seja representado por 24 bits, requer 3 Mb para seu armazenamento (sem compressão)
  - Um vídeo de 1 minuto, onde cada quadro seja descrito com imagens  $512 \times 512 \times 24$ , exibidos a uma taxa de 30 fps, requer 1.4 Gb de armazenamento sem compressão

# Introdução

## Representação e Descrição

- Estruturas adequadas de **representação** devem ser utilizadas para armazenar e manipular os objetos de interesse extraídos da imagem
- Imagens e vídeos requerem alta capacidade de armazenamento:
  - Uma imagem colorida  $1024 \times 1024$  pixels onde cada pixel seja representado por 24 bits, requer 3 Mb para seu armazenamento (sem compressão)
  - Um vídeo de 1 minuto, onde cada quadro seja descrito com imagens  $512 \times 512 \times 24$ , exibidos a uma taxa de 30 fps, requer 1.4 Gb de armazenamento sem compressão
- O processo de **Descrição** visa a extração de características ou propriedades que possam ser utilizadas na discriminação entre classes de objetos. Essas características são, em geral, descritas por atributos numéricos que formam um vetor de características

# Introdução

## Reconhecimento e Interpretação

- **Reconhecimento** ou **classificação** e o processo que atribui um identificador ou rótulo aos objetos da imagem, baseado nas características providas pelos seus descritores

# Introdução

## Reconhecimento e Interpretação

- **Reconhecimento** ou **classificação** é o processo que atribui um identificador ou rótulo aos objetos da imagem, baseado nas características providas pelos seus descritores
- O processo de **interpretação** consiste em atribuir um significado ao conjunto de objetos reconhecidos.

# Introdução

## Reconhecimento e Interpretação

- **Reconhecimento** ou **classificação** é o processo que atribui um identificador ou rótulo aos objetos da imagem, baseado nas características providas pelos seus descritores
- O processo de **interpretação** consiste em atribuir um significado ao conjunto de objetos reconhecidos.
- A forma dos contornos de células sanguíneas, por exemplo, pode auxiliar o diagnóstico de anemias por meio da contagem automática das células em uma amostra de sangue

# Python

# A Linguagem Python



# python

<https://www.python.org/downloads/>

# A Linguagem Python



# python

<https://www.python.org/downloads/>

Materiais de instalação: http:

[//www.joinville.udesc.br/portal/professores/alexandre/materiais/materiais.html](http://www.joinville.udesc.br/portal/professores/alexandre/materiais/materiais.html)

# A Linguagem Python

## Características

- Python é uma linguagem de programação e ambiente de prototipação de aprendizagem rápida

# A Linguagem Python

## Características

- Python é uma linguagem de programação e ambiente de prototipação de aprendizagem rápida
- Suporta vários paradigmas de programação: orientado a objetos, procedural e outros...

# A Linguagem Python

## Características

- Python é uma linguagem de programação e ambiente de prototipação de aprendizagem rápida
- Suporta vários paradigmas de programação: orientado a objetos, procedural e outros...
- Manipulação ágil de matrizes

# A Linguagem Python

## Características

- Python é uma linguagem de programação e ambiente de prototipação de aprendizagem rápida
- Suporta vários paradigmas de programação: orientado a objetos, procedural e outros...
- Manipulação ágil de matrizes
- Módulos sobre processamento matemático: SciPy, NumPy, ...

# A Linguagem Python

## Características

- Python é uma linguagem de programação e ambiente de prototipação de aprendizagem rápida
- Suporta vários paradigmas de programação: orientado a objetos, procedural e outros...
- Manipulação ágil de matrizes
- Módulos sobre processamento matemático: SciPy, NumPy, ...
- Módulos sobre manipulação de imagens: PIL, ia636, ...

# A Linguagem Python

## Características

- Python é uma linguagem de programação e ambiente de prototipação de aprendizagem rápida
- Suporta vários paradigmas de programação: orientado a objetos, procedural e outros...
- Manipulação ágil de matrizes
- Módulos sobre processamento matemático: SciPy, NumPy, ...
- Módulos sobre manipulação de imagens: PIL, ia636, ...
- Módulos sobre visualização gráfica: Matplotlib

# A Linguagem Python

## Características

- Python é uma linguagem de programação e ambiente de prototipação de aprendizagem rápida
- Suporta vários paradigmas de programação: orientado a objetos, procedural e outros...
- Manipulação ágil de matrizes
- Módulos sobre processamento matemático: SciPy, NumPy, ...
- Módulos sobre manipulação de imagens: PIL, ia636, ...
- Módulos sobre visualização gráfica: Matplotlib
- Farto materiais de consulta online (tutoriais)

# A Linguagem Python

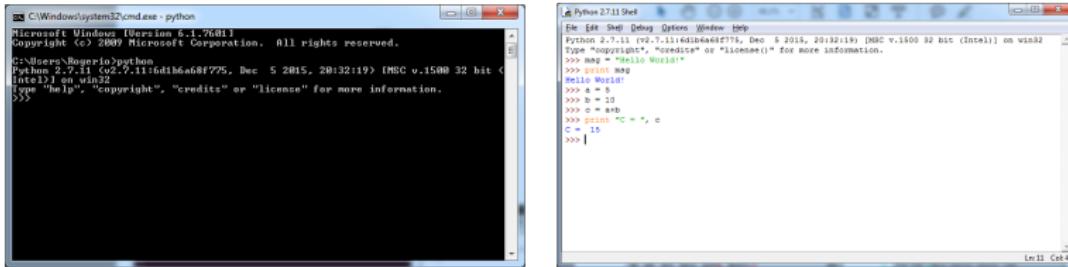
## Características

- Python é uma linguagem de programação e ambiente de prototipação de aprendizagem rápida
- Suporta vários paradigmas de programação: orientado a objetos, procedural e outros...
- Manipulação ágil de matrizes
- Módulos sobre processamento matemático: SciPy, NumPy, ...
- Módulos sobre manipulação de imagens: PIL, ia636, ...
- Módulos sobre visualização gráfica: Matplotlib
- Farto materiais de consulta online (tutoriais)
- Duas correntes de desenvolvimento: versão 2.X e 3.X (não são 100% compatíveis entre si) → adotaremos a versão 2.X

# A Linguagem Python

## Ambientes de Programação

### Terminal janela do interpretador Python IDLE Interactive DeveLopment Environment



C:\Windows\system32\cmd.exe - python

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Rogério>python
Python 2.7.11 (v2.7.11:6d1b6a6af37f, Dec 5 2015, 20:32:19) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Python 2.7.11 Shell

```
File Edit Shell Debug Options Window Help
Python 2.7.11 (v2.7.11:6d1b6a6af37f, Dec 5 2015, 20:32:19) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> msg = "Hello World!"
>>> print msg
Hello World!
>>> a = 5
>>> b = 10
>>> c = a+b
>>> print "%d + %d = %d" % (a, b, c)
15
>>>
```

# A Linguagem Python

## Exemplos de Programação (1)

```
1 x = 0
2 a , b = 1, 2
3 p1 = p2 = 1
4 tupla = (1,2,3) #tupla = variavel multivalorada
5
6 print "A = ", a, "B = " b
7 print tupla
8
9 x = input("Digite um valor:")
```

# A Linguagem Python

## Exemplos de Programação (2)

```
1 if a > 0 and b != -1:  
2     print "1"  
3 elif a <= 0:  
4     print "2"  
5 else:  
6     print "3"  
7  
8 for i in range(0,10):  
9     print i  
10  
11 while x > 0:  
12     ...
```

# A Linguagem Python

## Exemplos de Programação (3)

```
1 lista = [1,2,3,4,5]
2 listaHeterogenea = [1,3.5, "PIM", (0,0,0)]
3 matriz = [[1,2],[3,4],[5,6]]
4 print len(matriz)
5 print len(matriz[0])
6 print matriz
7
8 dicionario = {"A":0, "B":1, "C":2} #conjunto chave:valor
9 print dicionario ["B"]
10 dicionario ["D"] = 3 #insere novo elemento no dicionario
11 print dicionario
```

# A Linguagem Python

## Exemplos de Programação (4)

```
1 v1 = v2 = [4,1,0,-2,10,5]
2 v3 = v1 + v2 #concatenacao de listas
3 v3.sort()
4 print v3
5
6 # intercala e soma os elementos em v1 e v2
7 v4 = [x + y for x, y in zip(v1,v2)]
```

# A Linguagem Python

## Exemplos de Programação (5)

```
1 def sum(a,b):  
2     return a + b  
3  
4 a = 5  
5 b = 6  
6 c = sum(a, b)  
7 print c
```

# A Linguagem Python

## Módulos Externos: NumPy

- Matrizes e vetores são representados por objetos *array*

# A Linguagem Python

## Módulos Externos: NumPy

- Matrizes e vetores são representados por objetos *array*
- Imagens são matrizes numéricas

# A Linguagem Python

## Módulos Externos: NumPy

- Matrizes e vetores são representados por objetos *array*
- Imagens são matrizes numéricas
- **NumPy** é uma biblioteca para manipulação de *array*

# A Linguagem Python

## Módulos Externos: NumPy

- Matrizes e vetores são representados por objetos *array*
- Imagens são matrizes numéricas
- **NumPy** é uma biblioteca para manipulação de *array*
- Disponível em <http://www.numpy.org/>

# A Linguagem Python

## Módulos Externos: NumPy

- Matrizes e vetores são representados por objetos *array*
- Imagens são matrizes numéricas
- **NumPy** é uma biblioteca para manipulação de *array*
- Disponível em <http://www.numpy.org/>
- Tutoriais online:
  - [http://www.python-course.eu/matrix\\_arithmetic.php](http://www.python-course.eu/matrix_arithmetic.php)
  - [http://wiki.scipy.org/Tentative\\_NumPy\\_Tutorial](http://wiki.scipy.org/Tentative_NumPy_Tutorial)
  - [http://scipy-lectures.github.io/intro/numpy/array\\_object.html#indexing-and-slicing](http://scipy-lectures.github.io/intro/numpy/array_object.html#indexing-and-slicing)
  - [http://wiki.scipy.org/NumPy\\_for\\_Matlab\\_Users](http://wiki.scipy.org/NumPy_for_Matlab_Users)

# A Linguagem Python

## Módulos Externos: NumPy

- Para incluir o módulo:

```
import numpy
```

# A Linguagem Python

## Módulos Externos: NumPy

- Para incluir o módulo:

```
import numpy
```

- Para criar um array 1D:

```
vetor = numpy.array([1,2,3])
```

# A Linguagem Python

## Módulos Externos: NumPy

- Para incluir o módulo:

```
import numpy
```

- Para criar um array 1D:

```
vetor = numpy.array([1,2,3])
```

- Para criar um array 2D:

```
matriz = numpy.array([[1,2,3],[4,5,6]])
```

# A Linguagem Python

## Módulos Externos: NumPy

- Para incluir o módulo:

```
import numpy
```

- Para criar um array 1D:

```
vetor = numpy.array([1,2,3])
```

- Para criar um array 2D:

```
matriz = numpy.array([[1,2,3],[4,5,6]])
```

- Para criar um intervalo de valores

```
a = numpy.arange(0,10,2)
```

produz o vetor

```
array([0,2,4,6,8])
```

# A Linguagem Python

## Módulos Externos: NumPy

- Para incluir o módulo:

```
import numpy
```

- Para criar um array 1D:

```
vetor = numpy.array([1,2,3])
```

- Para criar um array 2D:

```
matriz = numpy.array([[1,2,3],[4,5,6]])
```

- Para criar um intervalo de valores

```
a = numpy.arange(0,10,2)
```

produz o vetor

```
array([0,2,4,6,8])
```

- O comando `numpy.zeros(2,5)` cria uma matriz  $2 \times 5$  de valores iguais a zero

```
array([[0.,0.,0.,0.,0.],[0.,0.,0.,0.,0.]])
```

# A Linguagem Python

## Módulos Externos: NumPy

- Para incluir o módulo:

```
import numpy
```

- Para criar um array 1D:

```
vetor = numpy.array([1,2,3])
```

- Para criar um array 2D:

```
matriz = numpy.array([[1,2,3],[4,5,6]])
```

- Para criar um intervalo de valores

```
a = numpy.arange(0,10,2)
```

produz o vetor

```
array([0,2,4,6,8])
```

- O comando `numpy.zeros(2,5)` cria uma matriz  $2 \times 5$  de valores iguais a zero

```
array([[0.,0.,0.,0.,0.],[0.,0.,0.,0.,0.]])
```

- A dimensão de um array é dada por

```
matriz = numpy.array([[1,2,3],[4,5,6]])  
matriz.shape
```

que resulta em  $(2,3)$

# A Linguagem Python

## Módulos Externos: NumPy

- Permite funções matemáticas como

```
a = numpy.sin(angle)
```

# A Linguagem Python

## Módulos Externos: NumPy

- Permite funções matemáticas como

```
a = numpy.sin(angle)
```

- Transposição de matriz:

```
matriz = numpy.array([[1,2,3],[4,5,6]])
matriz.T
```

# A Linguagem Python

## Módulos Externos: NumPy

- Permite funções matemáticas como

```
a = numpy.sin(angle)
```

- Transposição de matriz:

```
matriz = numpy.array([[1,2,3],[4,5,6]])
matriz.T
```

- Operações numéricas:

```
b = matriz * 2
```

# A Linguagem Python

## Módulos Externos: NumPy

- Permite funções matemáticas como

```
a = numpy.sin(angle)
```

- Transposição de matriz:

```
matriz = numpy.array([[1,2,3],[4,5,6]])
matriz.T
```

- Operações numéricas:

```
b = matriz * 2
```

- Multiplicação de matrizes:

```
mat1, mat2 = numpy.array([[1,2,3],[4,5,6]])
mat3 = mat1 * mat2
```

# A Linguagem Python

Módulos Externos: PIL

- **PIL** (*Python Image Library*) ferramenta para manipulação de imagens

# A Linguagem Python

## Módulos Externos: PIL

- **PIL** (*Python Image Library*) ferramenta para manipulação de imagens
- Disponível em <http://www.pythonware.com/products/pil>

# A Linguagem Python

## Módulos Externos: PIL

- **PIL** (*Python Image Library*) ferramenta para manipulação de imagens
- Disponível em <http://www.pythonware.com/products/pil>
- Tutoriais online:
  - <http://effbot.org/imagingbook/introduction.htm>
  - <http://www.geeks3d.com/20100930/tutorial-first-steps-with-pil-python-imaging-library/>

# A Linguagem Python

## Módulos Externos: PIL

- **PIL** (*Python Image Library*) ferramenta para manipulação de imagens
- Disponível em <http://www.pythonware.com/products/pil>
- Tutoriais online:
  - <http://effbot.org/imagingbook/introduction.htm>
  - <http://www.geeks3d.com/20100930/tutorial-first-steps-with-pil-python-imaging-library/>
- Exemplos de Uso:

```
1 from PIL import Image  
2 image = Image.open('python.png')  
3 image.show()  
4 r,g,b = image.getpixel( (x,y) )  
5 image.save('saida.png')
```

# A Linguagem Python

Módulos Externos: SciPy + Matplotlib

- **SciPy** ferramenta para manipulação de imagens

# A Linguagem Python

Módulos Externos: SciPy + Matplotlib

- **SciPy** ferramenta para manipulação de imagens
- Disponível em <https://www.scipy.org/>

# A Linguagem Python

Módulos Externos: SciPy + Matplotlib

- **SciPy** ferramenta para manipulação de imagens

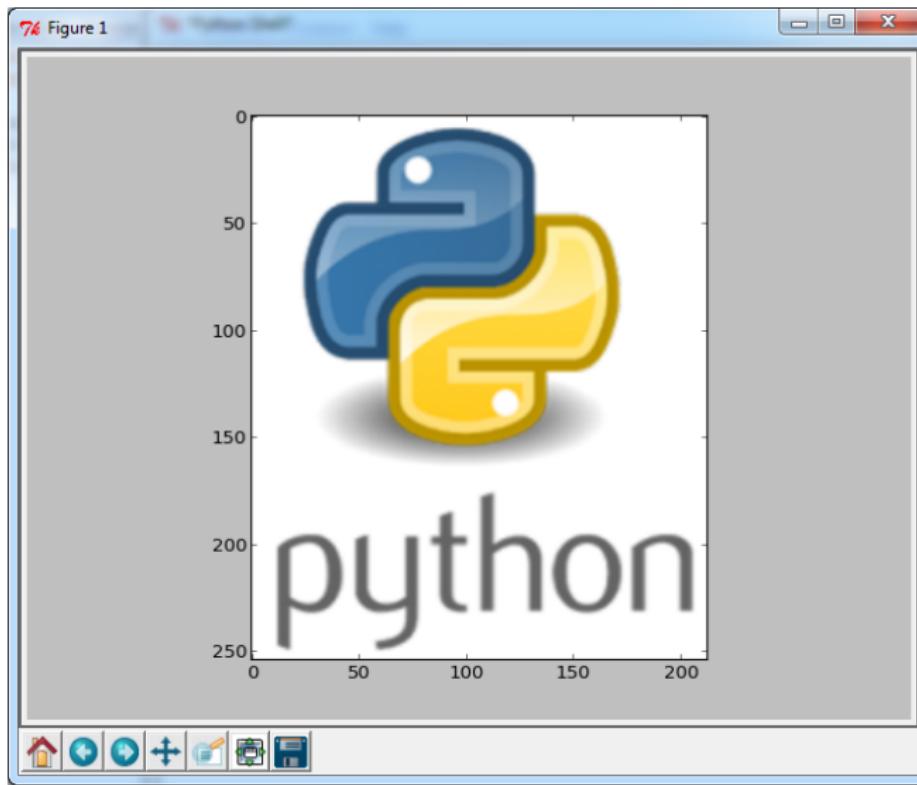
- Disponível em <https://www.scipy.org/>

- Exemplos de Uso:

```
1 import scipy.misc  
2 import matplotlib.pyplot as plt  
3 image = scipy.misc.imread('python.png')  
4 plt.imshow(image)  
5 plt.show()
```

# A Linguagem Python

Módulos Externos: SciPy + Matplotlib



# Fundamentos de Imagens Digitais

# Fundamentos de Imagens Digitais

## Tipos de Sinais

### Sinal

É a manifestação de um fenômeno que pode ser expresso de forma quantitativa

# Fundamentos de Imagens Digitais

## Tipos de Sinais

### Sinal

É a manifestação de um fenômeno que pode ser expresso de forma quantitativa

- É representado como uma função de uma ou mais variáveis independentes

# Fundamentos de Imagens Digitais

## Tipos de Sinais

### Sinal

É a manifestação de um fenômeno que pode ser expresso de forma quantitativa

- É representado como uma função de uma ou mais variáveis independentes
- Contém informações acerca da natureza ou comportamento do fenômeno físico sob consideração

# Fundamentos de Imagens Digitais

## Tipos de Sinais

### Sinal

É a manifestação de um fenômeno que pode ser expresso de forma quantitativa

- É representado como uma função de uma ou mais variáveis independentes
- Contém informações acerca da natureza ou comportamento do fenômeno físico sob consideração
- Exemplo:
  - Sinal de voz = função de uma variável (tempo)

# Fundamentos de Imagens Digitais

## Tipos de Sinais

### Sinal

É a manifestação de um fenômeno que pode ser expresso de forma quantitativa

- É representado como uma função de uma ou mais variáveis independentes
- Contém informações acerca da natureza ou comportamento do fenômeno físico sob consideração
- Exemplo:
  - Sinal de voz = função de uma variável (tempo)
  - Imagem = função de duas variáveis (espaço)

# Fundamentos de Imagens Digitais

## Tipos de Sinais

- Os sinais podem ser classificados em relação a como variam em função do tempo:

# Fundamentos de Imagens Digitais

## Tipos de Sinais

- Os sinais podem ser classificados em relação a como variam em função do tempo:

**Discreto** é definido por um conjunto de valores enumeráveis ou inteiros, cujo intervalo depende da natureza do sinal

# Fundamentos de Imagens Digitais

## Tipos de Sinais

- Os sinais podem ser classificados em relação a como variam em função do tempo:

**Discreto** é definido por um conjunto de valores enumeráveis ou inteiros, cujo intervalo depende da natureza do sinal

**Contínuo** seus estados podem ser definidos em qualquer instante de tempo, ou seja, sem interrupção

# Fundamentos de Imagens Digitais

## Tipos de Sinais

- Os sinais podem ser classificados em relação a como variam em função do tempo:

**Discreto** é definido por um conjunto de valores enumeráveis ou inteiros, cujo intervalo depende da natureza do sinal

**Contínuo** seus estados podem ser definidos em qualquer instante de tempo, ou seja, sem interrupção

- Um sinal ainda pode ser classificado como:

**Analógico** variações contínuas no tempo. Ex.: onda sonora

# Fundamentos de Imagens Digitais

## Tipos de Sinais

- Os sinais podem ser classificados em relação a como variam em função do tempo:

**Discreto** é definido por um conjunto de valores enumeráveis ou inteiros, cujo intervalo depende da natureza do sinal

**Contínuo** seus estados podem ser definidos em qualquer instante de tempo, ou seja, sem interrupção

- Um sinal ainda pode ser classificado como:

**Analógico** variações contínuas no tempo. Ex.: onda sonora

**Digital** variações discretas no tempo. Ex.: código Morse

# Fundamentos de Imagens Digitais

## Tipos de Sinais

- É possível converter um tipo de sinal em outro através de circuitos eletrônicos:

# Fundamentos de Imagens Digitais

## Tipos de Sinais

- É possível converter um tipo de sinal em outro através de circuitos eletrônicos:
- Exemplos:
  - Ondas sonoras (sinal analógico) podem ser capturadas por um microfone para serem manipuladas em um computador (sinal digital)

# Fundamentos de Imagens Digitais

## Tipos de Sinais

- É possível converter um tipo de sinal em outro através de circuitos eletrônicos:
- Exemplos:
  - Ondas sonoras (sinal analógico) podem ser capturadas por um microfone para serem manipuladas em um computador (sinal digital)
  - Em contrapartida, para se reproduzir o áudio nos auto-falantes do computador, o sinal digital deve ser convertido para analógico

# Fundamentos de Imagens Digitais

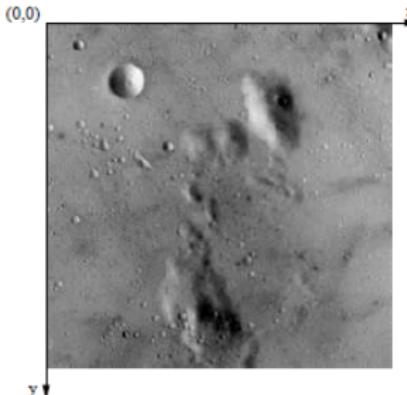
## Modelo de Imagens

- Uma imagem pode ser definida como uma função da intensidade luminosa  $f(x, y)$ , cujo valor ou amplitude nas coordenadas  $(x, y)$  denota a intensidade ou brilho da imagem naquele ponto

# Fundamentos de Imagens Digitais

## Modelo de Imagens

- Uma imagem pode ser definida como uma função da intensidade luminosa  $f(x, y)$ , cujo valor ou amplitude nas coordenadas  $(x, y)$  denota a intensidade ou brilho da imagem naquele ponto
- Convencionalmente o *pixel* no canto superior esquerdo indica a origem  $(0, 0)$  do sistema de coordenadas



# Fundamentos de Imagens Digitais

## Modelo de Imagens

**Illuminância** quantidade de luz incidente na cena  $\equiv i(x, y)$

Determinada pela fonte de luz. Unidade de medida:  
lumens/ $m^2$  (*lux*)

# Fundamentos de Imagens Digitais

## Modelo de Imagens

**Iluminância** quantidade de luz incidente na cena  $\equiv i(x, y)$

Determinada pela fonte de luz. Unidade de medida:  
lumens/m<sup>2</sup> (*lux*)

**Reflectância** quantidade de luz refletida na cena  $\equiv r(x, y)$

Determinada pela característica dos objetos na cena.  
Unidade de medida: valores percentuais [0 . . . 1]

# Fundamentos de Imagens Digitais

## Modelo de Imagens

**Iluminância** quantidade de luz incidente na cena  $\equiv i(x, y)$

Determinada pela fonte de luz. Unidade de medida:  
lumens/m<sup>2</sup> (*lux*)

**Reflectância** quantidade de luz refletida na cena  $\equiv r(x, y)$

Determinada pela característica dos objetos na cena.  
Unidade de medida: valores percentuais [0 . . . 1]

**Intensidade de luz** composição das duas componentes:

$$f(x, y) = i(x, y) \times r(x, y)$$

onde:  $0 < i(x, y) < \infty$  e  $0 < r(x, y) < 1$

# Fundamentos de Imagens Digitais

## Modelo de Imagens

Exemplos de  $i(x, y)$ :

Dia claro de Sol  $900.000 \text{ lumens}/m^2$

Dia nublado  $10.000 \text{ lumens}/m^2$

Interior de um escritório  $1.000 \text{ lumens}/m^2$

Noite com lua cheia  $0.1 \text{ lumens}/m^2$

# Fundamentos de Imagens Digitais

## Modelo de Imagens

Exemplos de  $i(x, y)$ :

Dia claro de Sol  $900.000 \text{ lumens}/m^2$

Dia nublado  $10.000 \text{ lumens}/m^2$

Interior de um escritório  $1.000 \text{ lumens}/m^2$

Noite com lua cheia  $0.1 \text{ lumens}/m^2$

Exemplos de  $r(x, y)$ :

Neve 0.93

Parede branca 0.80

Aço inoxidável 0.65

Veludo preto 0.01

# Fundamentos de Imagens Digitais

## Digitalização

### Digitalização

Processo computacional que permite produzir uma imagem digital.  
É o resultado de duas etapas: amostragem e quantização

# Fundamentos de Imagens Digitais

## Digitalização

### Digitalização

Processo computacional que permite produzir uma imagem digital.  
É o resultado de duas etapas: amostragem e quantização

Amostragem discretização do domínio da imagem nas direções  $x$  e  $y$ , gerando uma matriz  $M \times N$  amostras

# Fundamentos de Imagens Digitais

## Digitalização

### Digitalização

Processo computacional que permite produzir uma imagem digital.  
É o resultado de duas etapas: amostragem e quantização

**Amostragem** discretização do domínio da imagem nas direções  $x$  e  $y$ , gerando uma matriz  $M \times N$  amostras

**Quantização** escolher o número de níveis de cinza ( $L$ ) em uma imagem monocromática

# Fundamentos de Imagens Digitais

## Digitalização

- Um *pixel* é dado por cada elemento da matriz  $f(x, y)$  onde  $0 \leq x \leq M - 1$  e  $0 \leq y \leq N - 1$

# Fundamentos de Imagens Digitais

## Digitalização

- Um *pixel* é dado por cada elemento da matriz  $f(x, y)$  onde  $0 \leq x \leq M - 1$  e  $0 \leq y \leq N - 1$
- A imagem resultante é dado por uma matriz de tamanho  $M \times N$  :

$$f(x, y) \approx \begin{bmatrix} f(0, 0) & f(1, 0) & \dots & f(M - 1, 0) \\ f(0, 1) & f(1, 1) & \dots & f(M - 1, 1) \\ \vdots & \vdots & \ddots & \vdots \\ f(0, N - 1) & f(1, N - 1) & \dots & f(M - 1, N - 1) \end{bmatrix}$$

# Fundamentos de Imagens Digitais

## Digitalização

- Dimensão de um *pixel* denota o espaçamento físico entre as amostras

# Fundamentos de Imagens Digitais

## Digitalização

- Dimensão de um *pixel* denota o espaçamento físico entre as amostras
- Cada *pixel* está associado a uma **escala de cinza**, no intervalo  $[L_{min}, L_{max}]$  onde  $L_{min} \leq f(x, y) \leq L_{max}$

# Fundamentos de Imagens Digitais

## Digitalização

- Dimensão de um *pixel* denota o espaçamento físico entre as amostras
- Cada *pixel* está associado a uma **escala de cinza**, no intervalo  $[L_{min}, L_{max}]$  onde  $L_{min} \leq f(x, y) \leq L_{max}$
- Em uma imagem monocromática, cada intensidade  $f$  em  $(x, y)$  é denominado **nível de cinza**

# Fundamentos de Imagens Digitais

## Digitalização

- Dimensão de um *pixel* denota o espaçamento físico entre as amostras
- Cada *pixel* está associado a uma **escala de cinza**, no intervalo  $[L_{min}, L_{max}]$  onde  $L_{min} \leq f(x, y) \leq L_{max}$
- Em uma imagem monocromática, cada intensidade  $f$  em  $(x, y)$  é denominado **nível de cinza**
- Abordagem clássica: associar preto (0) para os tons escuros e branco (255) para os tons mais claros

# Fundamentos de Imagens Digitais

## Digitalização

- É usual considerar as dimensões da imagem e o número de níveis de cinza como potências de 2

# Fundamentos de Imagens Digitais

## Digitalização

- É usual considerar as dimensões da imagem e o número de níveis de cinza como potências de 2
- São chamadas **imagens binárias** àquelas com apenas 2 níveis de cinza

# Fundamentos de Imagens Digitais

## Digitalização

- É usual considerar as dimensões da imagem e o número de níveis de cinza como potências de 2
- São chamadas **imagens binárias** àquelas com apenas 2 níveis de cinza
- Imagens binárias são de grande importância: reduzem o espaço necessário para representação e permitem fácil manipulação (operações lógicas clássicas)

# Fundamentos de Imagens Digitais

## Resolução Espacial vs Profundidade de Imagem

Resolução Gráfica indica a densidade de pixels da imagem, ou seja, quanto maior a densidade menor o espaçamento da amostragem

# Fundamentos de Imagens Digitais

## Resolução Espacial vs Profundidade de Imagem

**Resolução Gráfica** indica a densidade de pixels da imagem, ou seja, quanto maior a densidade menor o espaçamento da amostragem

**Profundidade de Imagem** indica o número de níveis de quantização ( $b$ ) associado a cada pixel da imagem:  $L = 2^b$

# Fundamentos de Imagens Digitais

## Resolução Espacial vs Profundidade de Imagem

**Resolução Gráfica** indica a densidade de pixels da imagem, ou seja, quanto maior a densidade menor o espaçamento da amostragem

**Profundidade de Imagem** indica o número de níveis de quantização ( $b$ ) associado a cada pixel da imagem:  $L = 2^b$   
A profundidade representa a quantidade de bits necessários para armazenar cada pixel da imagem

# Fundamentos de Imagens Digitais

## Resolução Espacial vs Profundidade de Imagem

**Resolução Gráfica** indica a densidade de pixels da imagem, ou seja, quanto maior a densidade menor o espaçamento da amostragem

**Profundidade de Imagem** indica o número de níveis de quantização ( $b$ ) associado a cada pixel da imagem:  $L = 2^b$   
A profundidade representa a quantidade de bits necessários para armazenar cada pixel da imagem

### Exemplo

Para  $L = 256$  temos: cada pixel é representado por valores entre 0 e 255, ou seja,  $b = 8$  (bits) ou 1 byte

# Fundamentos de Imagens Digitais

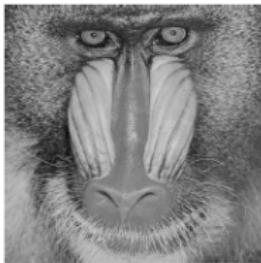
Resolução Espacial vs Profundidade de Imagem

M	N	L = 2	L = 8	L = 32	L = 128	L = 512
320	256	10240	30720	51200	71680	92160
480	320	19200	57600	96000	134400	172800
640	400	32000	96000	160000	224000	288000
800	600	60000	180000	300000	420000	540000
1024	720	92160	276480	460800	645120	829440
1280	800	128000	384000	640000	896000	1152000

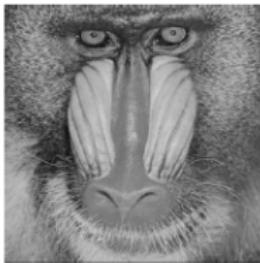
# Fundamentos de Imagens Digitais

Resolução Espacial vs Profundidade de Imagem

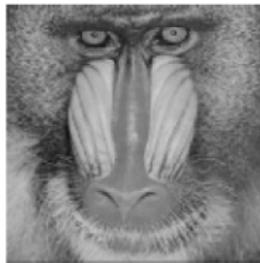
Exemplos de Resolução Gráfica



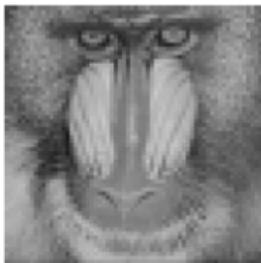
(a)  $512 \times 512$



(b)  $256 \times 256$



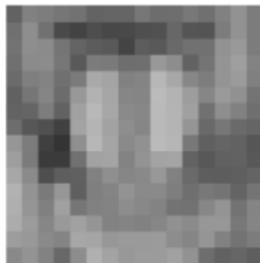
(c)  $128 \times 128$



(d)  $64 \times 64$



(e)  $32 \times 32$

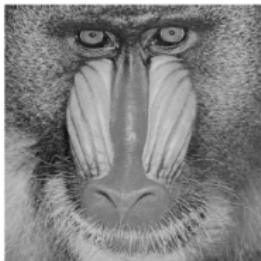


(f)  $16 \times 16$

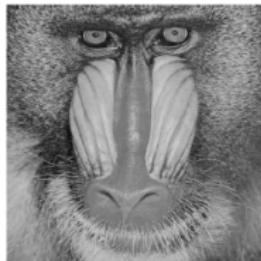
# Fundamentos de Imagens Digitais

Resolução Espacial vs Profundidade de Imagem

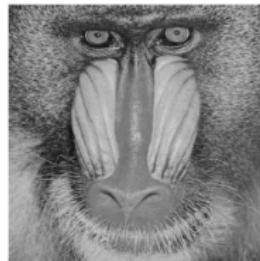
## Exemplos de Profundidades de Imagem



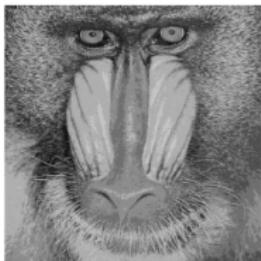
(a) 64



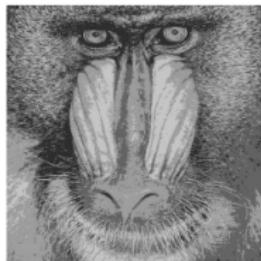
(b) 32



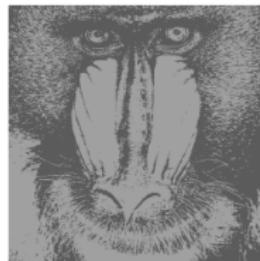
(c) 16



(d) 8



(e) 4



(f) 2

# Fundamentos de Imagens Digitais

## Representação de Imagens Digitais - Exercício

- Usando Scipy, extrair as informações de resolução gráfica de uma imagem:

# Fundamentos de Imagens Digitais

## Representação de Imagens Digitais - Exercício

- Usando Scipy, extrair as informações de resolução gráfica de uma imagem:

```
1 import scipy.misc  
2 import matplotlib.pyplot as plt  
3 path = "imagens/"  
4 image = scipy.misc.imread(path+'mandril.jpg')  
5 print image.shape
```

# Fundamentos de Imagens Digitais

## Representação de Imagens Digitais - Exercício

- Usando Scipy, extrair as informações de resolução gráfica de uma imagem:

```
1 import scipy.misc  
2 import matplotlib.pyplot as plt  
3 path = "imagens/"  
4 image = scipy.misc.imread(path+'mandril.jpg')  
5 print image.shape
```

Retorno: (256, 256, 3)

# Fundamentos de Imagens Digitais

## Representação de Imagens Digitais - Exercício

- Usando Scipy, extrair as informações de resolução gráfica de uma imagem:

```
1 import scipy.misc  
2 import matplotlib.pyplot as plt  
3 path = "imagens/"  
4 image = scipy.misc.imread(path+'mandril.jpg')  
5 print image.shape
```

Retorno: **(256, 256, 3)**

o que indica uma imagem com resolução gráfica  $256 \times 256$  pixels onde cada um tem 3 níveis de profundidade (RGB)

# Fundamentos de Imagens Digitais

## Representação de Imagens Digitais

```
1 import scipy.misc  
2 import matplotlib.pyplot as plt  
3 path = "imagens/"  
4 image = scipy.misc.imread(path+'mandril.jpg')  
5 print image
```

# Fundamentos de Imagens Digitais

## Representação de Imagens Digitais

```
1 import scipy.misc
2 import matplotlib.pyplot as plt
3 path = "imagens/"
4 image = scipy.misc.imread(path+'mandril.jpg')
5 print image
```

```
Python Shell
File Edit Options Windows Help
Python 2.7.3 (default, Apr 10 2012, 23:31:26) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
==== No Subprocess ====
[[118 88 23]
 [88 88 0]
 [159 118 89]
 ...
 [145 142 98]
 [188 137 70]
 [84 82 31]

 [[128 95 93]
 [176 147 81]
 [112 87 0]
 ...
 [98 96 37]
 [95 95 31]
 [105 105 43]

 ...
 [[ 88 129 123]
 [134 174 173]
 [148 187 194]
 ...
 [106 107 93]
 [103 101 104]
 [106 102 116]

 [[108 146 140]
 [141 181 181]
 [130 168 179]
 ...
 ...]]
```

# Fundamentos de Imagens Digitais

## Representação de Imagens Digitais

- Matrizes são simples de armazenar, manipular e visualizar os dados representados

# Fundamentos de Imagens Digitais

## Representação de Imagens Digitais

- Matrizes são simples de armazenar, manipular e visualizar os dados representados
- A desvantagem é a invariabilidade espacial, ou seja, não é adaptativa a irregularidades na imagem  $\Rightarrow$  redundância de dados

# Fundamentos de Imagens Digitais

## Representação de Imagens Digitais

- Matrizes são simples de armazenar, manipular e visualizar os dados representados
- A desvantagem é a invariabilidade espacial, ou seja, não é adaptativa a irregularidades na imagem  $\Rightarrow$  redundância de dados
- Como solução, utilizam-se modelos de compressão de imagem  $\Rightarrow$  redução do espaço de armazenamento e do tempo de transmissão

# Fundamentos de Imagens Digitais

## Representação de Imagens Digitais

- Pode-se otimizar a representação de uma imagem utilizando-se múltiplas resoluções (resoluções hierárquicas) ⇒ representação em pirâmide

# Fundamentos de Imagens Digitais

## Representação de Imagens Digitais

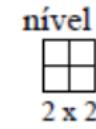
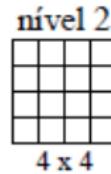
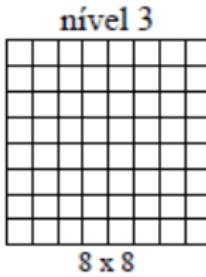
- Pode-se otimizar a representação de uma imagem utilizando-se múltiplas resoluções (resoluções hierárquicas)  $\Rightarrow$  representação em pirâmide
- Uma imagem produz uma série de matrizes (níveis) com resoluções diferentes. Assim, para representar uma matriz  $N \times N$  (onde  $N$  é uma potência de 2) com  $k$  versões temos:

$$1 \times 1, \quad \frac{N}{2} \times \frac{N}{2}, \quad \frac{N}{4} \times \frac{N}{4}, \quad \frac{N}{8} \times \frac{N}{8}, \dots$$

# Fundamentos de Imagens Digitais

## Representação de Imagens Digitais

Exemplo de Representação hierárquica da Matriz de uma Imagem



# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais

- Uma imagem **multibanda** ou **multiespectral** é um tipo de representação hierárquica onde, para cada pixel são armazenados múltiplos níveis de cor  $f(x, y) = (L_1, L_2, \dots, L_N)$  onde cada  $L_{\max} \leq L_i \leq L_{\min}$  para cada  $i = 1, 2, 3, \dots, N$

# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais

- Uma imagem **multibanda** ou **multiespectral** é um tipo de representação hierárquica onde, para cada pixel são armazenados múltiplos níveis de cor  $f(x, y) = (L_1, L_2, \dots, L_N)$  onde cada  $L_{max} \leq L_i \leq L_{min}$  para cada  $i = 1, 2, 3, \dots, N$
- Exemplos de imagem multibanda: representação de diferentes grandezas como temperatura, pressão ou frequência

# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais

- Uma imagem **multibanda** ou **multiespectral** é um tipo de representação hierárquica onde, para cada pixel são armazenados múltiplos níveis de cor  $f(x, y) = (L_1, L_2, \dots, L_N)$  onde cada  $L_{max} \leq L_i \leq L_{min}$  para cada  $i = 1, 2, 3, \dots, N$
- Exemplos de imagem multibanda: representação de diferentes grandezas como temperatura, pressão ou frequência
- Uma imagem multiespectral pode ser composta de várias imagens monocromáticas

# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais

- Uma imagem **multibanda** ou **multiespectral** é um tipo de representação hierárquica onde, para cada pixel são armazenados múltiplos níveis de cor  $f(x, y) = (L_1, L_2, \dots, L_N)$  onde cada  $L_{max} \leq L_i \leq L_{min}$  para cada  $i = 1, 2, 3, \dots, N$
- Exemplos de imagem multibanda: representação de diferentes grandezas como temperatura, pressão ou frequência
- Uma imagem multiespectral pode ser composta de várias imagens monocromáticas
- Aplicações: sensoriamento remoto que operam em diferentes faixas do espectro eletromagnético

# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais

- Imagens coloridas são exemplos de imagens multibandas onde cada ponto  $(x, y)$  é definido por:

# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais

- Imagens coloridas são exemplos de imagens multibandas onde cada ponto  $(x, y)$  é definido por:  
**Luminância** quantidade de brilho do ponto

# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais

- Imagens coloridas são exemplos de imagens multibandas onde cada ponto  $(x, y)$  é definido por:
  - Luminância quantidade de brilho do ponto
  - Matiz comprimento de onda dominante

# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais

- Imagens coloridas são exemplos de imagens multibandas onde cada ponto  $(x, y)$  é definido por:
  - Luminância quantidade de brilho do ponto
  - Matiz comprimento de onda dominante
  - Saturação grau de pureza da matiz

# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais

- Imagens coloridas são exemplos de imagens multibandas onde cada ponto  $(x, y)$  é definido por:
  - Luminância quantidade de brilho do ponto
  - Matiz comprimento de onda dominante
  - Saturação grau de pureza da matiz
- Uma representação comum de cor utiliza três bandas primárias de cor: vermelha (R), verde (G) e azul (B), com profundidade de 1 byte de cor (24 bits/pixel)

# Fundamentos de Imagens Digitais

Imagens Multibandas ou Multiespectrais

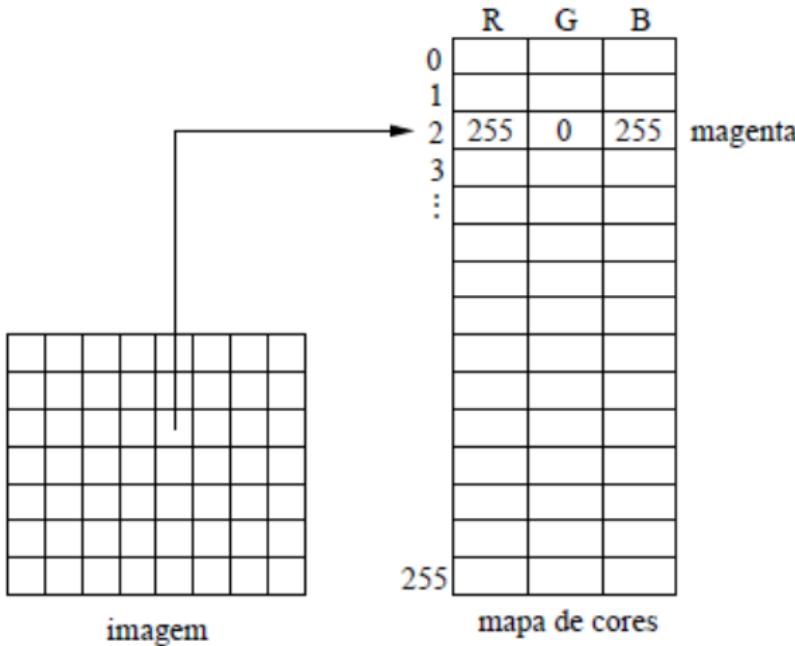
Representação Multibanda RGB de uma Imagem



# Fundamentos de Imagens Digitais

Imagens Multibandas ou Multiespectrais

Representação Multibanda RGB de uma Imagem por Tabela de Cores



# Fundamentos de Imagens Digitais

Imagens Multibandas ou Multiespectrais

Exemplo de Representação da Matriz de uma Imagem

120	138	120	151	139
110	129	129	139	146
150	138	137	138	129
137	129	129	128	137
146	145	131	132	145



# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais

```
1 import scipy . misc  
2  
3 path = "imagens/"  
4 data = [ # 20 linhas com 20 pixels cada uma  
5     [[0,0,0],[0,0,0],...,[0,0,0],[0,0,0]],      #0,0,0 = PRETO  
6     [[0,0,0],[0,0,0],...,[0,0,0],[0,0,0]],  
7     [[255,255,255],[255,255,255],...,[255,255,255]],    #255, 255, 255 = BRANCO  
8     [[255,255,255],[255,255,255],...,[255,255,255]],  
9     ...  
10    ]  
11    scipy . misc.imsave(path+"listras . png",data)
```

# Fundamentos de Imagens Digitais

Imagens Multibandas ou Multiespectrais



# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais - Exercício

- Desenvolver um programa `xadrez.py`, que crie uma imagem de resolução  $W \times H$  como um tabuleiro de xadrez onde o tamanho do quadro é  $T$
- Exemplo de uso: `GeraTabuleiro(256, 256, 64, "xadrez.png")`

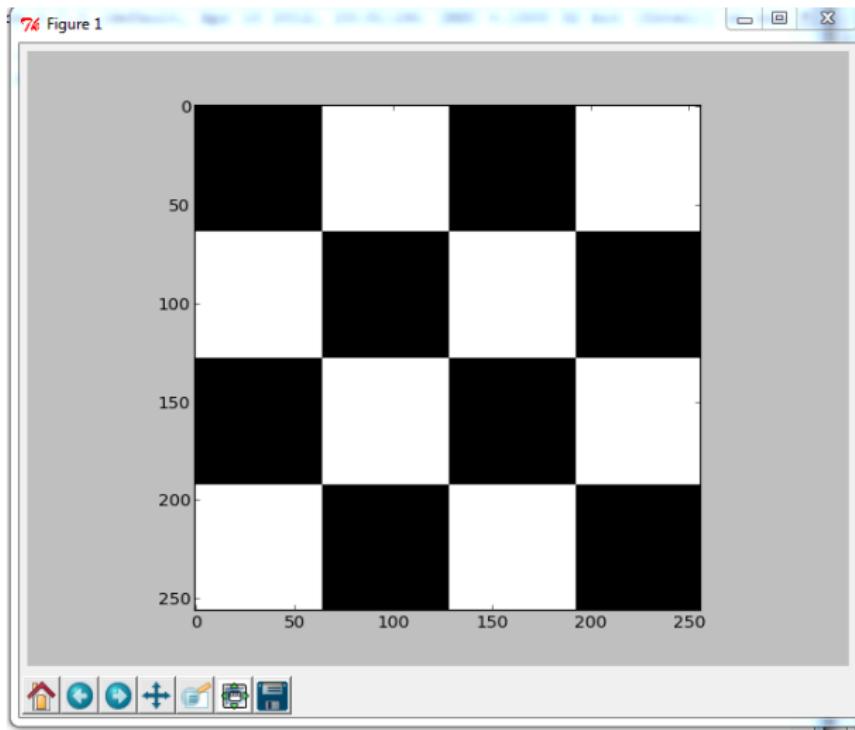
# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais - Exercício

```
1 import scipy.misc
2 import matplotlib.pyplot as plt
3
4 path = "imagens/"
5
6 def GeraTabuleiro(W, H, T, filename):
7     data = []
8     for i in range(0,H):
9         row = []
10        for j in range(0,W):
11            lvl = 0
12            if (j/T) % 2 == 0:
13                if (i/T) % 2 == 0:
14                    lvl = 0
15                else:
16                    lvl = 255
17            else:
18                if (i/T) % 2 == 0:
19                    lvl = 255
20                else:
21                    lvl = 0
22            row.append([lvl , lvl , lvl ])
23        data.append(row)
24
25    scipy.misc.imsave(filename,data)
26
27 # =====
28
29 GeraTabuleiro(256, 256, 64, path+"xadrez.png")
30
31 resultado = scipy.misc.imread(path+"xadrez.png")
32 plt.imshow(resultado)
33 plt.show()
```

# Fundamentos de Imagens Digitais

Imagens Multibandas ou Multiespectrais - Exercício



# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais - Exercício

- Desenvolver um programa `piramide.py` que, dados a imagem original e um fator de amostragem  $T$ , crie uma imagem que seja a amostragem de cada  $T$  pixels da imagem original, ou seja, calcule um nível específico da pirâmide de amostragem para uma imagem
- Exemplo de uso: `GeraNivel('lena.jpg', 8, 'piramide.png')`

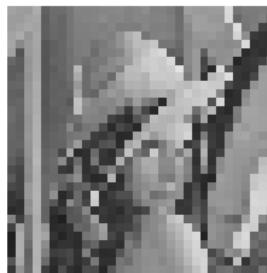
# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais - Exercício

```
1 import scipy.misc
2 import matplotlib.pyplot as plt
3
4 path = "imagens/"
5
6 def GeraNivel(input, T, output):
7
8     imagemIN = scipy.misc.imread(input)
9     W,H = imagemIN.shape
10
11    data = []
12    for i in range(0,H,T):
13        row = []
14        for j in range(0,W,T):
15            lvl = imagemIN[i][j]
16            row.append([lvl, lvl, lvl])
17        data.append(row)
18
19    scipy.misc.imsave(output,data)
20
21 # =====
22
23 input = path + "lena.jpg"
24 output = path + "piramide.png"
25
26 GeraNivel(input, 8, output)
27
28 resultado = scipy.misc.imread(output)
29 plt.imshow(resultado)
30 plt.show()
```

# Fundamentos de Imagens Digitais

Imagens Multibandas ou Multiespectrais - Exercício



# Fundamentos de Imagens Digitais

## Imagens Multibandas ou Multiespectrais - Exercício

- Adaptar o programa anterior para que crie uma imagem cuja amostragem seja a média de cada área  $T \times T$  pixels da imagem original

# Fundamentos de Imagens Digitais

## Imagens Multidimensionais

### Definição

Uma imagem multidimensional 3D é representada como uma sequência de imagens monocromáticas ou multibandas ao longo de um eixo dimensional  $z$  ou temporal  $t$

# Fundamentos de Imagens Digitais

## Imagens Multidimensionais

### Definição

Uma imagem multidimensional 3D é representada como uma sequência de imagens monocromáticas ou multibandas ao longo de um eixo dimensional  $z$  ou temporal  $t$

- Nas tomografias computadorizadas realizam uma sequência de fatias de varredura (cortes), geralmente uniformemente variadas no espaço, assim produzindo uma imagem multimensional (volumétrica)

# Fundamentos de Imagens Digitais

## Imagens Multidimensionais

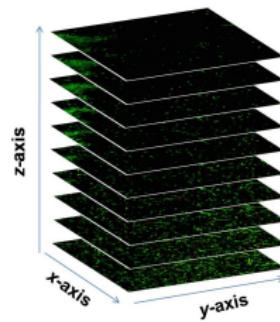
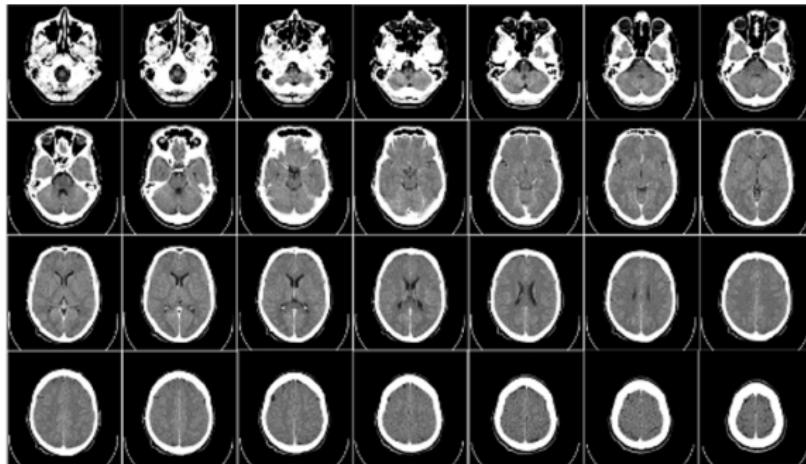
### Definição

Uma imagem multidimensional 3D é representada como uma sequência de imagens monocromáticas ou multibandas ao longo de um eixo dimensional  $z$  ou temporal  $t$

- Nas tomografias computadorizadas realizam uma sequência de fatias de varredura (cortes), geralmente uniformemente variadas no espaço, assim produzindo uma imagem multidimensional (volumétrica)
- Nestas imagens, um pixel é representado por suas dimensões  $p \times p$  e o espaçamento  $d$  nos cortes. Desta forma, um pixel 3D é dado por  $p \times p \times d$  e é denominado **voxel** (*volumetric pixel*)

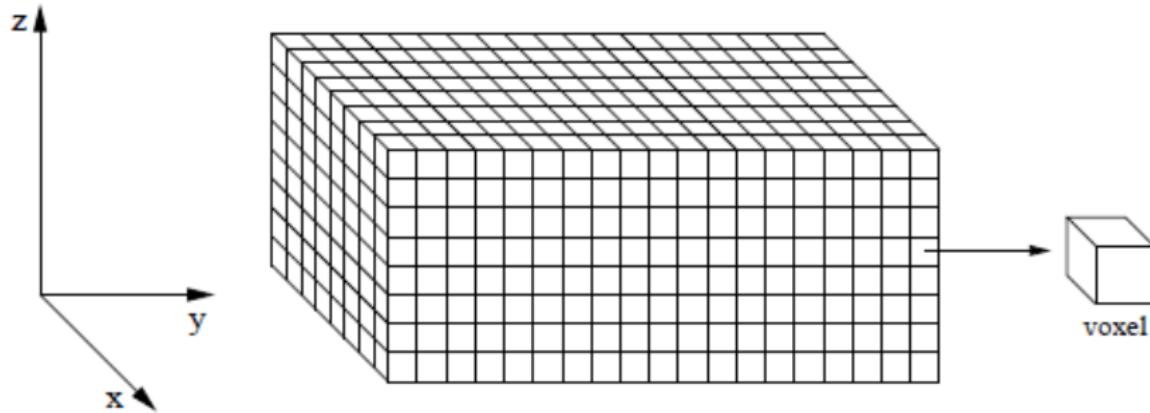
# Fundamentos de Imagens Digitais

## Imagens Multidimensionais



# Fundamentos de Imagens Digitais

## Imagens Multidimensionais



# Fundamentos de Imagens Digitais

## Sistema Visual Humano

- Visão é um dos principais sentidos humanos e envolve funções complexas como detecção, localização, reconhecimento e interpretação do ambiente

# Fundamentos de Imagens Digitais

## Sistema Visual Humano

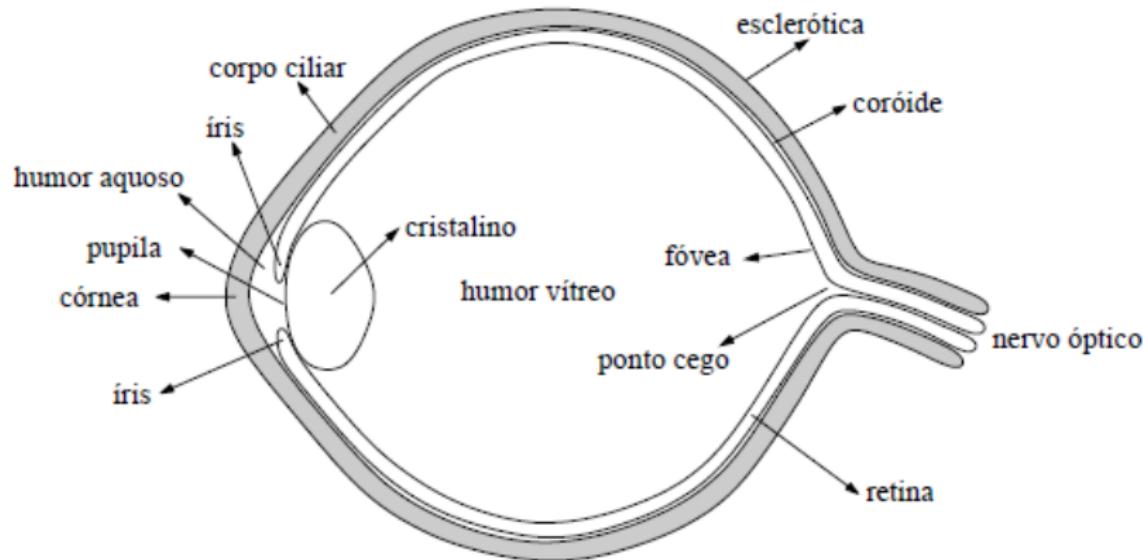
- Visão é um dos principais sentidos humanos e envolve funções complexas como detecção, localização, reconhecimento e interpretação do ambiente

### Visão Computacional

É a área da computação que visa dotar as máquinas com capacidades visuais inspirada no funcionamento do sistema visual humano, tais como: adquirir, analisar e interpretar informações visuais

# Fundamentos de Imagens Digitais

## Sistema Visual Humano



# Fundamentos de Imagens Digitais

## Sistema Visual Humano

Córnea funciona como uma lente cujo poder de refração permite a formação de uma imagem sobre a retina

# Fundamentos de Imagens Digitais

## Sistema Visual Humano

Córnea funciona como uma lente cujo poder de refração permite a formação de uma imagem sobre a retina

Íris e Pupila é a abertura frontal variável do olho que controla a quantidade de luz que entra no olho

# Fundamentos de Imagens Digitais

## Sistema Visual Humano

Córnea funciona como uma lente cujo poder de refração permite a formação de uma imagem sobre a retina

Íris e Pupila é a abertura frontal variável do olho que controla a quantidade de luz que entra no olho

Cristalino lente gelatinosa e elástica que auxilia a córnea na focalização da luz, ajustando a visão para objetos próximos e distantes

# Fundamentos de Imagens Digitais

## Sistema Visual Humano

Córnea funciona como uma lente cujo poder de refração permite a formação de uma imagem sobre a retina

Íris e Pupila é a abertura frontal variável do olho que controla a quantidade de luz que entra no olho

Cristalino lente gelatinosa e elástica que auxilia a córnea na focalização da luz, ajustando a visão para objetos próximos e distantes

Retina camada de tecidos nervosos composta por células sensíveis à luz:

# Fundamentos de Imagens Digitais

## Sistema Visual Humano

Córnea funciona como uma lente cujo poder de refração permite a formação de uma imagem sobre a retina

Íris e Pupila é a abertura frontal variável do olho que controla a quantidade de luz que entra no olho

Cristalino lente gelatinosa e elástica que auxilia a córnea na focalização da luz, ajustando a visão para objetos próximos e distantes

Retina camada de tecidos nervosos composta por células sensíveis à luz:

Cones células sensíveis à cor

Bastonetes células responsáveis por perceber a intensidade de energia luminosa do ambiente (claro/escuro)

# Fundamentos de Imagens Digitais

Sensores digitais como uma Metáfora para o Sistema Visual Humano

- Olho Humano ⇒ Câmera Fotográfica = princípio de funcionamento

# Fundamentos de Imagens Digitais

Sensores digitais como uma Metáfora para o Sistema Visual Humano

- Olho Humano ⇒ Câmera Fotográfica = princípio de funcionamento
- Tampa da lente ⇒ pálpebra do olho = abertura para entrada de luz

# Fundamentos de Imagens Digitais

Sensores digitais como uma Metáfora para o Sistema Visual Humano

- Olho Humano ⇒ Câmera Fotográfica = princípio de funcionamento
- Tampa da lente ⇒ pálpebra do olho = abertura para entrada de luz
- Obturador da câmera ⇒ íris/pupila do olho = controle da quantidade de luz incidente

# Fundamentos de Imagens Digitais

Sensores digitais como uma Metáfora para o Sistema Visual Humano

- Olho Humano ⇒ Câmera Fotográfica = princípio de funcionamento
- Tampa da lente ⇒ pálpebra do olho = abertura para entrada de luz
- Obturador da câmera ⇒ íris/pupila do olho = controle da quantidade de luz incidente
- Lentes da câmera ⇒ córnea & cristalino = focalização

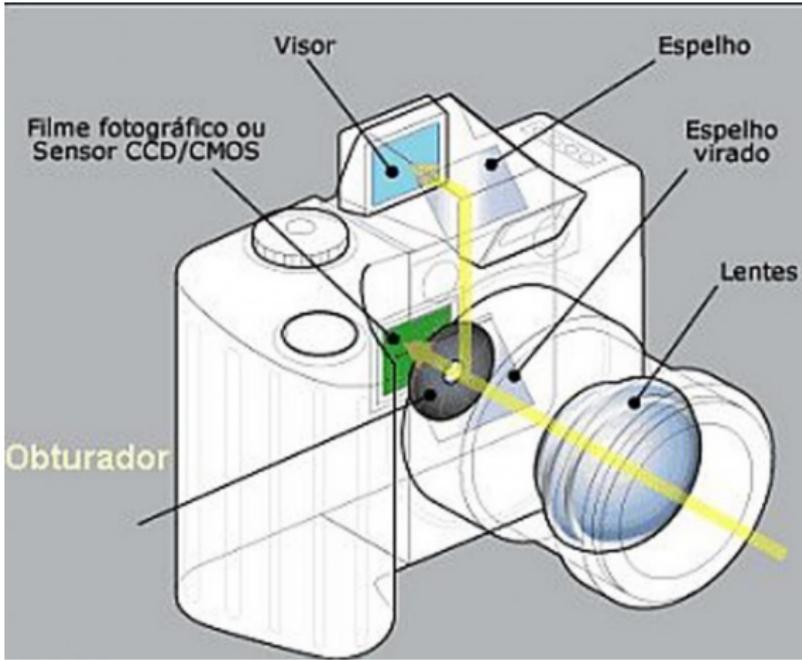
# Fundamentos de Imagens Digitais

Sensores digitais como uma Metáfora para o Sistema Visual Humano

- Olho Humano ⇒ Câmera Fotográfica = princípio de funcionamento
- Tampa da lente ⇒ pálpebra do olho = abertura para entrada de luz
- Obturador da câmera ⇒ íris/pupila do olho = controle da quantidade de luz incidente
- Lentes da câmera ⇒ córnea & cristalino = focalização
- Filmes fotográficos e/ou sensor CCD ⇒ retina = captação e armazenamento da informação ótica

# Fundamentos de Imagens Digitais

Sensores digitais como uma Metáfora para o Sistema Visual Humano



<http://www.gazetadopovo.com.br/blogs/foto-sintese/cursode-fotografia/>

# Fundamentos de Imagens Digitais

## Ruídos em Imagens

### Ruído

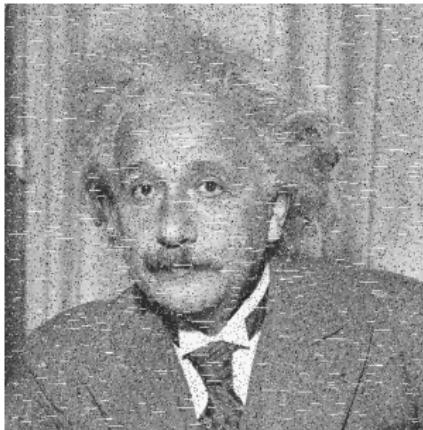
É uma degradação inerente a muitos processos de captura, transmissão e/ou processamento de imagens

# Fundamentos de Imagens Digitais

## Ruídos em Imagens

### Ruído

É uma degradação inerente a muitos processos de captura, transmissão e/ou processamento de imagens



<http://www.vision.ime.usp.br/~nina/cursos/mac5746-03/pratica.html>

# Fundamentos de Imagens Digitais

## Ruídos em Imagens

- Pode ser modelado como uma variável aleatória  $z$ , caracterizada por uma função de probabilidade  $p(z)$

# Fundamentos de Imagens Digitais

## Ruídos em Imagens

- Pode ser modelado como uma variável aleatória  $z$ , caracterizada por uma função de probabilidade  $p(z)$
- Tipos mais comuns de ruído:
  - Impulsivo
  - Gaussiano
  - Uniforme
  - Erlang
  - Exponencial
  - Rayleigh
  - Poisson

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: impulsivo

- Caracteriza-se pela ocorrência de *pixels* cujos valores de luminosidade diferem significativamente de seus vizinhos

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: impulsivo

- Caracteriza-se pela ocorrência de *pixels* cujos valores de luminosidade diferem significativamente de seus vizinhos
- Tipo comum: ruído **sal-e-pimenta** = pixels brancos e pretos

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: impulsivo

- Caracteriza-se pela ocorrência de *pixels* cujos valores de luminosidade diferem significativamente de seus vizinhos
- Tipo comum: ruído **sal-e-pimenta** = pixels brancos e pretos

$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & \text{caso contrário} \end{cases}$$

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: impulsivo

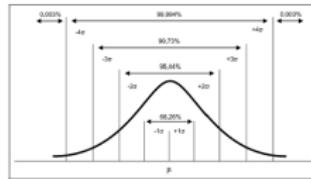


<https://maalencar.wordpress.com/2012/05/17/bconvolution-examples-with-matlab/>

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: gaussiano

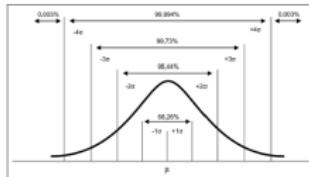
- Caracteriza-se pela ocorrência de *pixels* cujos valores variam de intensidade de acordo com uma distribuição gaussiana



# Fundamentos de Imagens Digitais

## Ruídos em Imagens: gaussiano

- Caracteriza-se pela ocorrência de *pixels* cujos valores variam de intensidade de acordo com uma distribuição gaussiana

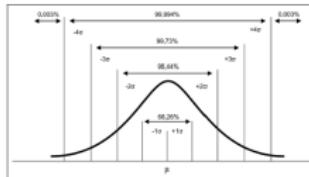


- Usado para modelar degradação proveniente da aquisição digital de imagens

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: gaussiano

- Caracteriza-se pela ocorrência de *pixels* cujos valores variam de intensidade de acordo com uma distribuição gaussiana



- Usado para modelar degradação proveniente da aquisição digital de imagens

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(z-\mu)^2}{2\sigma^2}}$$

$\mu$  = média e  $\sigma^2$  = variância da variável aleatória  $z$

# Fundamentos de Imagens Digitais

Ruídos em Imagens: gaussiano



<http://www.dca.fee.unicamp.br/~lotufo/Courses/ia-636-1995/mylene/proj.html>

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: uniforme

- Também chamado de **ruído de quantização** é originado do processo de conversão analógico-digital

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: uniforme

- Também chamado de **ruído de quantização** é originado do processo de conversão analógico-digital
- Apresenta uma distribuição uniforme ou retangular (distribuição aleatória)

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: uniforme

- Também chamado de **ruído de quantização** é originado do processo de conversão analógico-digital
- Apresenta uma distribuição uniforme ou retangular (distribuição aleatória)

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{para } a \leq z \leq b \\ 0 & \text{caso contrário} \end{cases}$$

onde:  $\mu = \frac{a+b}{2}$  e  $\sigma^2 = \frac{(b-a)^2}{12}$

# Fundamentos de Imagens Digitais

Ruídos em Imagens: uniforme



<http://www.dca.fee.unicamp.br/~lotufo/Courses/ia-636-1995/mylene/proj.html>

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: Erlang

- Também chamado de **ruído gama** é aquela que apresenta uma distribuição da forma:

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{para } z \geq 0 \\ 0 & \text{para } z < 0 \end{cases}$$

onde:  $a > 0$  e  $b$  é um valor inteiro positivo.

E ainda:

$$\mu = \frac{b}{a} \quad \text{e} \quad \sigma^2 = \frac{b}{a^2}$$

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: exponencial

- É um caso particular de ruído Erlang onde  $b = 1$ :

$$p(z) = \begin{cases} ae^{-az} & \text{para } z \geq 0 \\ 0 & \text{para } z < 0 \end{cases}$$

onde:  $a > 0$  e

$$\mu = \frac{1}{a} \quad \text{e} \quad \sigma^2 = \frac{1}{a^2}$$

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: exponencial



<http://www.dca.fee.unicamp.br/~lotufo/Courses/ia-636-1995/mylene/proj.html>

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: Rayleigh

- O ruído Rayleigh segue a seguinte distribuição de probabilidade:

$$p(z) = \begin{cases} \frac{2}{b}(z - a)e^{-\frac{(z-a)^2}{b}} & \text{para } z \geq a \\ 0 & \text{para } z < a \end{cases}$$

onde:

$$\mu = a + \sqrt{\pi \frac{b}{4}} \quad e \quad \sigma^2 = \frac{b(4 - \pi)}{4}$$

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: Rayleigh



<http://www.dca.fee.unicamp.br/~lotufo/Courses/ia-636-1995/mylene/proj.html>

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: Poisson

- O ruído Rayleigh segue a seguinte distribuição de probabilidade:

$$p(z) = \begin{cases} \frac{e^{-\mu} \mu^z}{z!} & \text{para } z \geq 0 \\ 0 & \text{para } z < 0 \end{cases}$$

onde:  $\mu$  = média da função de probabilidade e a variância  $\sigma^2 = \mu$

# Fundamentos de Imagens Digitais

## Ruídos em Imagens: Poisson



<http://www.dca.fee.unicamp.br/~lotufo/Courses/ia-636-1995/mylene/proj.html>

# Fundamentos de Imagens Digitais

## Entropia

### Entropia ou Incerteza

É uma medida da quantidade de informação transferida por um canal ou gerada por uma fonte

Foi proposta por Shannon (1948)

# Fundamentos de Imagens Digitais

## Entropia

### Entropia ou Incerteza

É uma medida da quantidade de informação transferida por um canal ou gerada por uma fonte

Foi proposta por Shannon (1948)

- Quanto maior o valor da entropia, mais incerteza e portanto mais informação estará associada ao canal

# Fundamentos de Imagens Digitais

## Entropia

### Entropia ou Incerteza

É uma medida da quantidade de informação transferida por um canal ou gerada por uma fonte

Foi proposta por Shannon (1948)

- Quanto maior o valor da entropia, mais incerteza e portanto mais informação estará associada ao canal
- O princípio fundamental da teoria de informação estabelece que a geração de informação pode ser modelada como um processo probabilístico

# Fundamentos de Imagens Digitais

## Entropia

### Entropia ou Incerteza

É uma medida da quantidade de informação transferida por um canal ou gerada por uma fonte

Foi proposta por Shannon (1948)

- Quanto maior o valor da entropia, mais incerteza e portanto mais informação estará associada ao canal
- O princípio fundamental da teoria de informação estabelece que a geração de informação pode ser modelada como um processo probabilístico
- Uma imagem pode ser considerada como o resultado de um processo aleatório, no qual a probabilidade  $p_i$  corresponde à probabilidade de um pixel em uma imagem digital assumir um valor de intensidade  $i$ ,  $i = 0, 1, \dots, L_{max}$

# Fundamentos de Imagens Digitais

## Entropia

- A distribuição dos níveis de intensidade da imagem pode ser transformada em uma função densidade de probabilidade, dividindo-se o número de pixels de intensidade  $i$ , denotado  $n_i$ , pelo número total  $n$  de pixels na imagem:

$$p_i = \frac{n_i}{n}$$

onde:

$$\sum_{i=0}^{L_{max}} p_i = 1$$

# Fundamentos de Imagens Digitais

## Entropia

- A distribuição dos níveis de intensidade da imagem pode ser transformada em uma função densidade de probabilidade, dividindo-se o número de pixels de intensidade  $i$ , denotado  $n_i$ , pelo número total  $n$  de pixels na imagem:

$$p_i = \frac{n_i}{n}$$

onde:

$$\sum_{i=0}^{L_{max}} p_i = 1$$

- A entropia  $H$  de uma imagem pode ser calculada por

$$H = - \sum_{i=0}^{L_{max}} p_i \log p_i$$

# Fundamentos de Imagens Digitais

## Entropia

- A entropia de uma imagem é uma medida positiva e, quando a base do logaritmo for dois, a unidade resultante é dada em bits

# Fundamentos de Imagens Digitais

## Entropia

- A entropia de uma imagem é uma medida positiva e, quando a base do logaritmo for dois, a unidade resultante é dada em bits
- O menor valor para a entropia é zero, ocorrendo quando todos os pixels possuem uma mesma intensidade de cinza

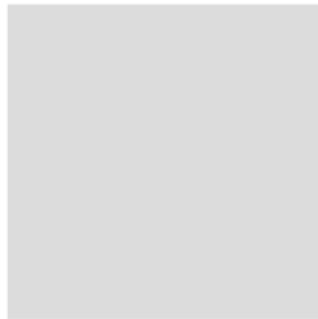
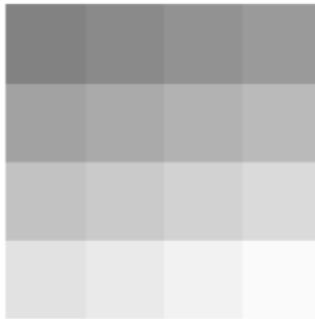
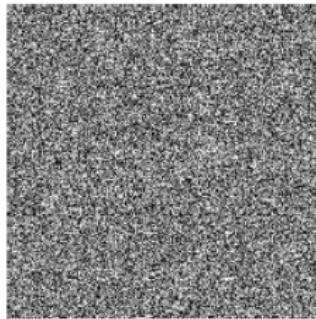
# Fundamentos de Imagens Digitais

## Entropia

- A entropia de uma imagem é uma medida positiva e, quando a base do logaritmo for dois, a unidade resultante é dada em bits
- O menor valor para a entropia é zero, ocorrendo quando todos os pixels possuem uma mesma intensidade de cinza
- Por outro lado, a máxima entropia ocorre quando uma imagem contém a mesma quantidade de pixels para todas as intensidades

# Fundamentos de Imagens Digitais

## Entropia

(a)  $H=0$ (b)  $H=1$ (c)  $H=1$ (d)  $H=2$ (e)  $H=4$ (f)  $H=8$

# Fundamentos de Imagens Digitais

Relacionamento entre Elementos de uma Imagem

Elementos:

Pixel elemento  $f$  em uma matriz bidimensional denotado por  
 $f(x, y)$

# Fundamentos de Imagens Digitais

Relacionamento entre Elementos de uma Imagem

Elementos:

Pixel elemento  $f$  em uma matriz bidimensional denotado por  
 $f(x, y)$

Voxel elemento  $f$  em uma matriz tridimensional denotado por  
 $f(x, y, z)$

# Fundamentos de Imagens Digitais

Relacionamento entre Elementos de uma Imagem

## Elementos:

**Pixel** elemento  $f$  em uma matriz bidimensional denotado por  $f(x, y)$

**Voxel** elemento  $f$  em uma matriz tridimensional denotado por  $f(x, y, z)$

## Relacionamento entre elementos:

- Vizinhança
- Conectividade
- Adjacência
- Caminho
- Componentes Conexos
- Borda e Interior

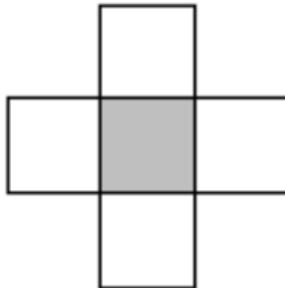
# Fundamentos de Imagens Digitais

## Vizinhança

### Vizinhança-4

Definida pelos quatro pixels vizinhos (horizontal e verticalmente)

$$(x - 1, y); (x + 1, y); (x, y - 1); (x, y + 1)$$

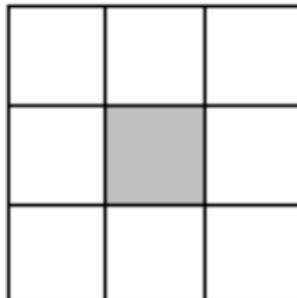


# Fundamentos de Imagens Digitais

## Vizinhança

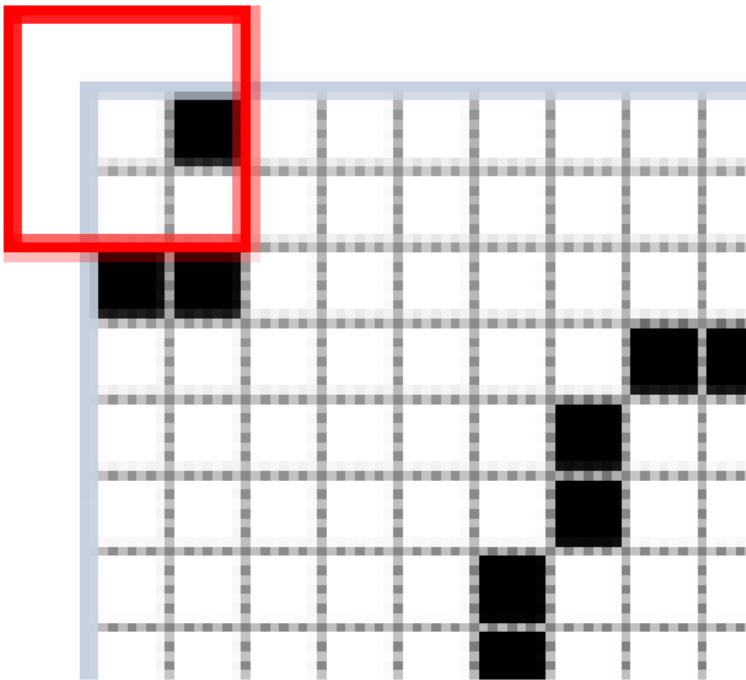
### Vizinhança-8

Definida pelos oito pixels vizinhos (horizontal e vertical e diagonalmente)

$$(x - 1, y); (x + 1, y); (x, y - 1); (x, y + 1)$$
$$(x - 1, y - 1); (x + 1, y + 1); (x - 1, y + 1); (x + 1, y - 1)$$


# Fundamentos de Imagens Digitais

## Vizinhança - Tratamento de bordas



# Fundamentos de Imagens Digitais

## Vizinhança - Tratamento de bordas

Três abordagens:

**Extensão de borda** Um pixel extra de valor nulo é adicionado a todo o contorno da matriz da imagem

# Fundamentos de Imagens Digitais

## Vizinhança - Tratamento de bordas

Três abordagens:

**Extensão de borda** Um pixel extra de valor nulo é adicionado a todo o contorno da matriz da imagem

Para uma imagem cujos índices variam entre  $0 \dots (M - 1)$  e  $0 \dots (N - 1)$ , a matriz passa a ter  $(M + 1) \times (N + 1)$  pixels

# Fundamentos de Imagens Digitais

## Vizinhança - Tratamento de bordas

Três abordagens:

**Extensão de borda** Um pixel extra de valor nulo é adicionado a todo o contorno da matriz da imagem

Para uma imagem cujos índices variam entre  $0 \dots (M - 1)$  e  $0 \dots (N - 1)$ , a matriz passa a ter  $(M + 1) \times (N + 1)$  pixels

**Reflexão de borda** Ao invés de aumentar o tamanho da imagem, essa abordagem reflete as margens opostas da imagem (matriz circular)  $\Rightarrow (-1, 0) \equiv (M, 0)$

# Fundamentos de Imagens Digitais

## Vizinhança - Tratamento de bordas

Três abordagens:

**Extensão de borda** Um pixel extra de valor nulo é adicionado a todo o contorno da matriz da imagem

Para uma imagem cujos índices variam entre  $0 \dots (M - 1)$  e  $0 \dots (N - 1)$ , a matriz passa a ter  $(M + 1) \times (N + 1)$  pixels

**Reflexão de borda** Ao invés de aumentar o tamanho da imagem, essa abordagem reflete as margens opostas da imagem (matriz circular)  $\Rightarrow (-1, 0) \equiv (M, 0)$

**Redução de Máscara** desconsiderar os pixels que estejam fora da área da imagem, o que exige testes condicionais extra para o processamento da máscara da vizinhança = mais lento

# Fundamentos de Imagens Digitais

## Vizinhança - Exercício

- Crie uma função em Python que leia uma imagem binária e conte a quantidade de pixels pretos isolados que a imagem possui em função de um dado critério de vizinhança
- Um ponto  $(x, y)$  é dito ser isolado se todos os seus vizinhos tiverem valor diferente do ponto

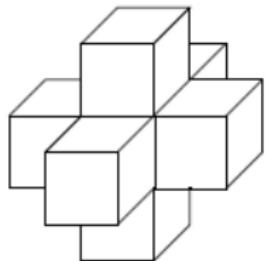
# Fundamentos de Imagens Digitais

## Vizinhança - Resolução do Exercício

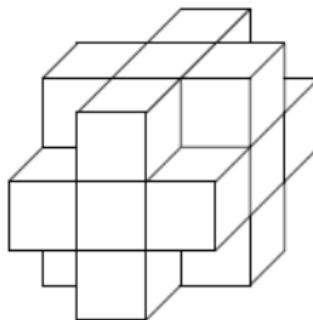
```
1 import scipy.misc
2 import numpy as np
3
4 path = "imagens/"
5
6 def TestaPixelIsolado (matriz,L,C):
7     if matriz[L][C][0] != 0:
8         return False
9     if L-1 >= 0:
10        if matriz[L-1][C][0] != 255:
11            return False
12        if L+1 < matriz.shape[0]:
13            if matriz[L+1][C][0] != 255:
14                return False
15        if C-1 >= 0:
16            if matriz[L][C-1][0] != 255:
17                return False
18        if C+1 < matriz.shape[1]:
19            if matriz[L][C+1][0] != 255:
20                return False
21    return True
22
23 def ContaPixel4( inputFile , outputFile ):
24     image = scipy.misc.imread( inputFile )
25     size = image.shape
26     npixels = 0
27     for L in range(0, size [0]):
28         for C in range(0, size [1]):
29             if TestaPixelIsolado (image,L,C):
30                 npixels = npixels + 1
31                 image[L][C] = [255,0,0] #substitui o pixel por vermelho
32     scipy .misc.imsave(outputFile ,image)
33     return npixels
34
35 print "Foram encontrados ", ContaPixel4(path+"teste.png", path+"resultado.png"), " pixels isolados "
```

# Fundamentos de Imagens Digitais

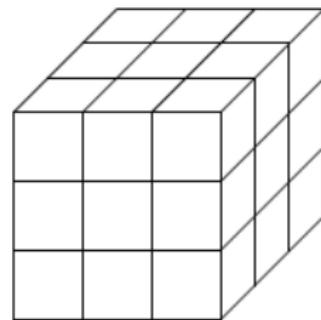
## Vizinhança em Voxels



(a) vizinhança-6



(b) vizinhança-18



(c) vizinhança-26

# Fundamentos de Imagens Digitais

## Conectividade

### Conectividade

Dois elementos são conexos se forem vizinhos e ainda apresentem **similaridade** (p.ex.: intensidade de cinza, cor ou textura)

# Fundamentos de Imagens Digitais

## Conectividade

### Conectividade

Dois elementos são conexos se forem vizinhos e ainda apresentem **similaridade** (p.ex.: intensidade de cinza, cor ou textura)

- Exemplo: Em uma imagem binária, dois pixels serão conexos se forem vizinhos e se tiverem o mesmo valor (0 ou 1)

# Fundamentos de Imagens Digitais

## Adjacência

- Um elemento  $f_1$  será adjacente a um elemento  $f_2$  se estes forem conexos de acordo com o critério de vizinhança adotado

# Fundamentos de Imagens Digitais

## Adjacência

- Um elemento  $f_1$  será adjacente a um elemento  $f_2$  se estes forem conexos de acordo com o critério de vizinhança adotado
- Dois subconjuntos  $S_1$  e  $S_2$  serão adjacentes se pelo menos um elemento de  $S_1$  for adjacente a algum elemento de  $S_2$

# Fundamentos de Imagens Digitais

## Caminho

Caminho:  $(x_1, y_1) \rightarrow (x_n, y_n)$

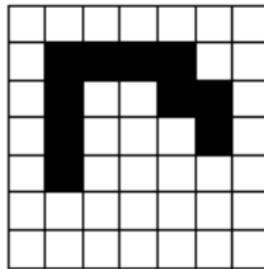
É uma sequência de pixels adjacentes distintos de coordenadas  $(x_1, y_1); (x_2, y_2); \dots; (x_n, y_n)$  onde  $n$  é o comprimento do caminho

# Fundamentos de Imagens Digitais

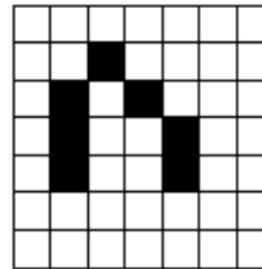
## Caminho

Caminho:  $(x_1, y_1) \rightarrow (x_n, y_n)$

É uma sequência de pixels adjacentes distintos de coordenadas  $(x_1, y_1); (x_2, y_2); \dots; (x_n, y_n)$  onde  $n$  é o comprimento do caminho



(a) caminho-4



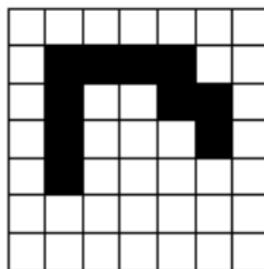
(b) caminho-8

# Fundamentos de Imagens Digitais

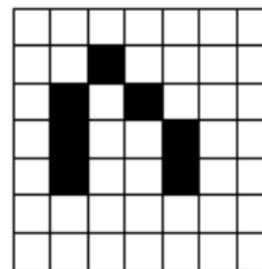
## Caminho

Caminho:  $(x_1, y_1) \rightarrow (x_n, y_n)$

É uma sequência de pixels adjacentes distintos de coordenadas  $(x_1, y_1); (x_2, y_2); \dots; (x_n, y_n)$  onde  $n$  é o comprimento do caminho



(a) caminho-4



(b) caminho-8

- O caminho-4 tem comprimento 10
- O caminho-8 tem comprimento 7
- O conceito de caminho é extensível também para voxels em 3D

# Fundamentos de Imagens Digitais

## Componentes Conexos

### Componentes Conexos

Qualquer subconjunto de elementos  $C$  que são conexos entre si

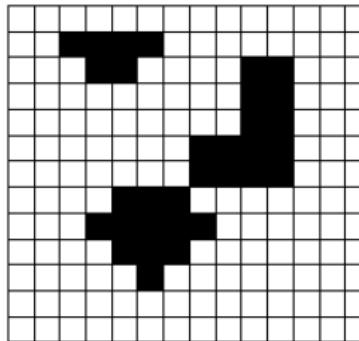
# Fundamentos de Imagens Digitais

## Componentes Conexos

### Componentes Conexos

Qualquer subconjunto de elementos  $C$  que são conexos entre si

- Quantos componentes conexos há na imagem abaixo ?



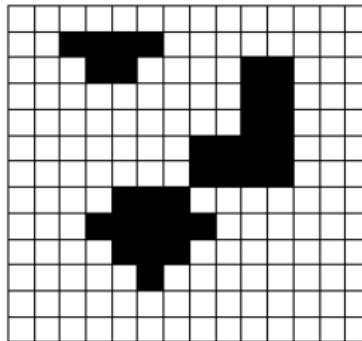
# Fundamentos de Imagens Digitais

## Componentes Conexos

### Componentes Conexos

Qualquer subconjunto de elementos  $C$  que são conexos entre si

- Quantos componentes conexos há na imagem abaixo ?



Vizinhança-4 3 componentes conexos

Vizinhança-8 2 componentes conexos

# Fundamentos de Imagens Digitais

## Borda e Interior

### Borda

É o conjunto de pixels pertencentes a um componente conexo  $S$  e que possuem vizinhança-4 com um ou mais pixels externos a  $S =$  contorno do componente  $S$

# Fundamentos de Imagens Digitais

## Borda e Interior

### Borda

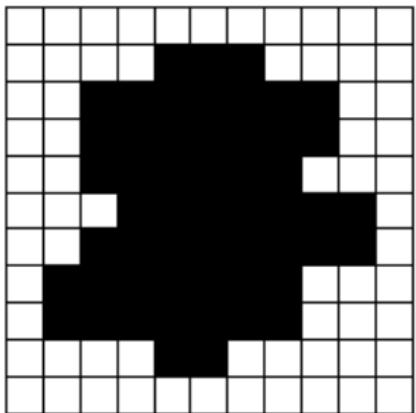
É o conjunto de pixels pertencentes a um componente conexo  $S$  e que possuem vizinhança-4 com um ou mais pixels externos a  $S =$  contorno do componente  $S$

### Interior

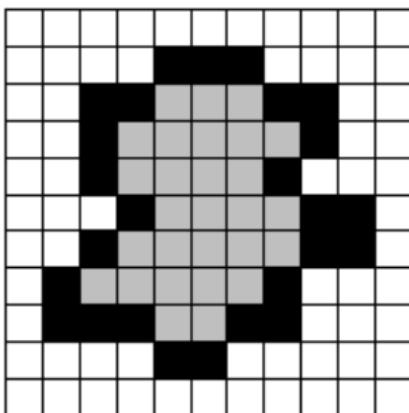
É o conjunto de pixels pertencentes ao componente  $S$  e que não estejam na sua borda

# Fundamentos de Imagens Digitais

## Borda e Interior



(a) imagem original



(b) pixels da borda e interior

■ borda  
■ interior

# Fundamentos de Imagens Digitais

## Medidas de Distância

- Existem diversas maneiras de se medir distância entre elementos em uma imagem → convenção matemática

# Fundamentos de Imagens Digitais

## Medidas de Distância

- Existem diversas maneiras de se medir distância entre elementos em uma imagem → convenção matemática
- Dados três elementos  $f_1, f_2$  e  $f_3$  com coordenadas  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ , qualquer métrica de distância  $D$  deve apresentar as seguintes propriedades:
  - ➊  $D(f_1, f_2) \leq 0$  ( $D(f_1, f_2) = 0$  se e somente se  $f_1 = f_2$ )

# Fundamentos de Imagens Digitais

## Medidas de Distância

- Existem diversas maneiras de se medir distância entre elementos em uma imagem → convenção matemática
- Dados três elementos  $f_1, f_2$  e  $f_3$  com coordenadas  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ , qualquer métrica de distância  $D$  deve apresentar as seguintes propriedades:
  - ➊  $D(f_1, f_2) \leq 0$  ( $D(f_1, f_2) = 0$  se e somente se  $f_1 = f_2$ )
  - ➋  $D(f_1, f_2) = D(f_2, f_1)$

# Fundamentos de Imagens Digitais

## Medidas de Distância

- Existem diversas maneiras de se medir distância entre elementos em uma imagem → convenção matemática
- Dados três elementos  $f_1, f_2$  e  $f_3$  com coordenadas  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ , qualquer métrica de distância  $D$  deve apresentar as seguintes propriedades:
  - ➊  $D(f_1, f_2) \leq 0$  ( $D(f_1, f_2) = 0$  se e somente se  $f_1 = f_2$ )
  - ➋  $D(f_1, f_2) = D(f_2, f_1)$
  - ➌  $D(f_1, f_3) \leq D(f_1, f_2) + D(f_2, f_3)$

# Fundamentos de Imagens Digitais

## Distância Euclidiana

### Distância Euclidiana

$$D_E(f_1, f_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

# Fundamentos de Imagens Digitais

## Distância Euclidiana

### Distância Euclidiana

$$D_E(f_1, f_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- Considerando um elemento central de coordenadas  $(x, y)$ , abaixo é apresentado o conjunto de distâncias  $D_E \leq 3$ :

			3			
	$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$	
	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	
3	2	1	0	1	2	3
	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	
	$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$	
			3			

# Fundamentos de Imagens Digitais

## Distância Euclidiana

### Distância Euclidiana

$$D_E(f_1, f_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- Considerando um elemento central de coordenadas  $(x, y)$ , abaixo é apresentado o conjunto de distâncias  $D_E \leq 3$ :

			3			
	$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$	
	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	
3	2	1	0	1	2	3
	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	
	$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$	
			3			

- A distância Euclidiana está mais próxima do caso contínuo, entretanto, requer mais esforço computacional e pode produzir valores fracionários

# Fundamentos de Imagens Digitais

Distância  $D_4$

Distância  $D_4$  ou *city-block*

$$D_4(f_1, f_2) = |x_1 - x_2| + |y_1 - y_2|$$

# Fundamentos de Imagens Digitais

Distância  $D_4$

Distância  $D_4$  ou *city-block*

$$D_4(f_1, f_2) = |x_1 - x_2| + |y_1 - y_2|$$

- Considerando um elemento central de coordenadas  $(x, y)$ , abaixo é apresentado o conjunto de distâncias  $D_4 \leq 3$ :

						3
			3	2	3	
		3	2	1	2	3
3	2	1	0	1	2	3
	3	2	1	2	3	
		3	2	3		
			3			

# Fundamentos de Imagens Digitais

Distância  $D_8$

Distância  $D_8$  ou *chessboard*

$$D_8(f_1, f_2) = \max(|x_1 - x_2|, |y_1 - y_2|)$$

# Fundamentos de Imagens Digitais

Distância  $D_8$

Distância  $D_8$  ou *chessboard*

$$D_8(f_1, f_2) = \max(|x_1 - x_2|, |y_1 - y_2|)$$

- Considerando um elemento central de coordenadas  $(x, y)$ , abaixo é apresentado o conjunto de distâncias  $D_8 \leq 3$ :

3	3	3	3	3	3	3
3	2	2	2	2	2	3
3	2	1	1	1	2	3
3	2	1	0	1	2	3
3	2	1	1	1	2	3
3	2	2	2	2	2	3
3	3	3	3	3	3	3

# Fundamentos de Imagens Digitais

## Operações Aritméticas

- Adição

$$\min(f_1(x, y) + f_2(x, y), 255)$$

- Subtração

$$\max(f_1(x, y) - f_2(x, y), 0)$$

- Multiplicação

$$\text{int}(\max(f_1(x, y) \times \epsilon, 255))$$

- Divisão<sup>1</sup>

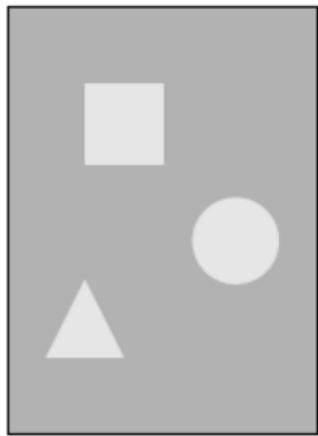
$$\text{int}\left(\frac{f_1(x, y)}{f_2(x, y)} \times 255\right)$$

---

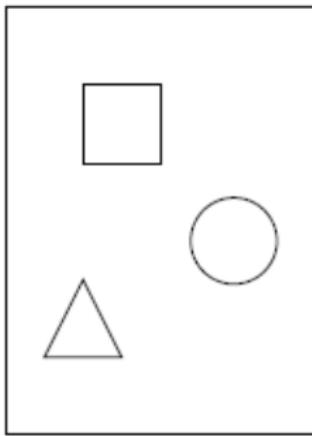
<sup>1</sup>Prevenir divisão por zero

# Fundamentos de Imagens Digitais

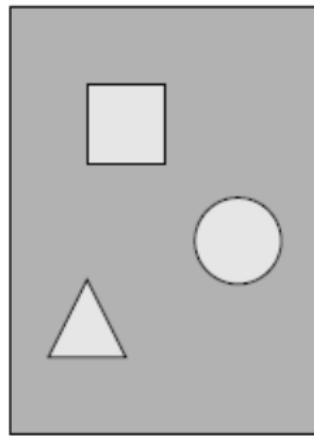
## Operações Aritméticas



(A) imagem original



(B) mapa de bordas



Adição: (A) + (B)

# Fundamentos de Imagens Digitais

## Operações Aritméticas

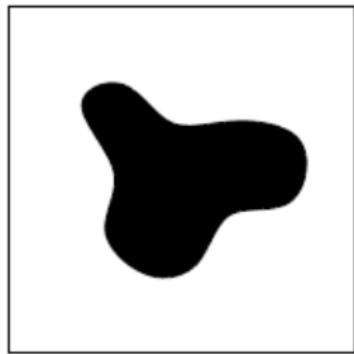


Imagen (A)

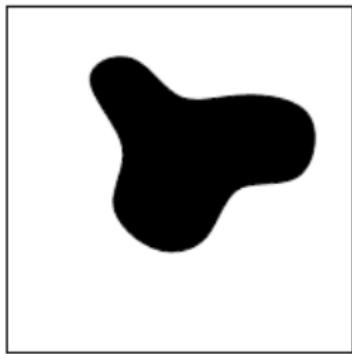


Imagen (B)



Diferença: (A) - (B)

# Fundamentos de Imagens Digitais

## Operações Aritméticas



Imagen original (A)

$\times 3 =$



Multiplicação:  $(A) * 3$

# Fundamentos de Imagens Digitais

## Operações Aritméticas

- Crie um programa que modifique os pixels de uma imagem por um fator  $\epsilon$

# Fundamentos de Imagens Digitais

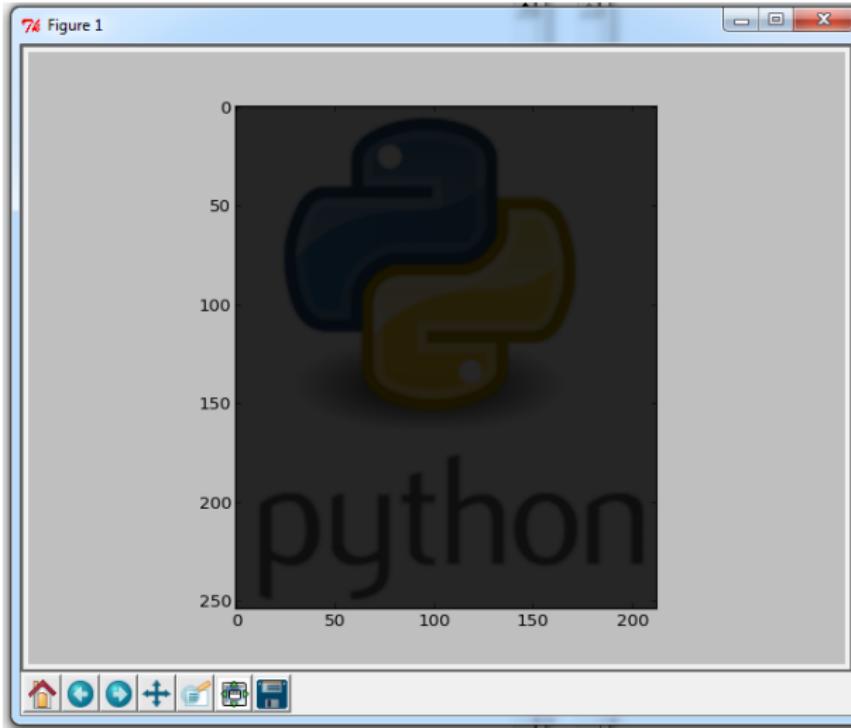
## Operações Aritméticas

- Crie um programa que modifique os pixels de uma imagem por um fator  $\epsilon$

```
1 import scipy.misc  
2 import matplotlib.pyplot as plt  
3 import numpy as np  
4  
5 path = "imagens/"  
6 logo = scipy.misc.imread(path+"python.png")  
7  
8 logo1 = np.array(logo*0.15, dtype=np.uint8)  
9  
10 plt.imshow(logo1)  
11 plt.show()
```

# Fundamentos de Imagens Digitais

## Operações Aritméticas



# Fundamentos de Imagens Digitais

## Operações Lógicas

- São aplicáveis apenas a imagens binárias onde preto ( $= 0$ ) indica objetos e branco ( $= 1$ ) indica fundo da imagem

# Fundamentos de Imagens Digitais

## Operações Lógicas

- São aplicáveis apenas a imagens binárias onde preto ( $= 0$ ) indica objetos e branco ( $= 1$ ) indica fundo da imagem

AND  $f_1(x, y)$  AND  $f_2(x, y)$

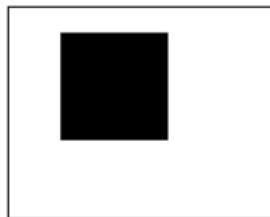
OR  $f_1(x, y)$  OR  $f_2(x, y)$

XOR  $f_1(x, y)$  XOR  $f_2(x, y)$

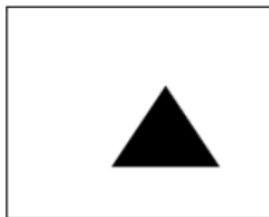
NOT NOT( $f_1(x, y)$ )

# Fundamentos de Imagens Digitais

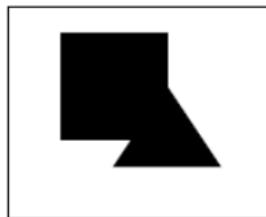
## Operações Lógicas



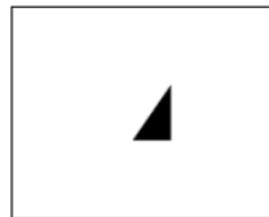
A



B



A OR B



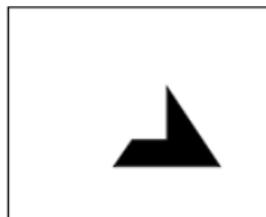
A AND B



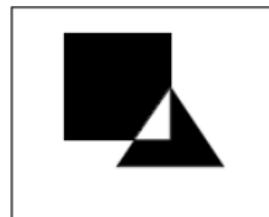
NOT(A)



NOT(B)



NOT(A) AND B



A XOR B

# Fundamentos de Imagens Digitais

## Operações Lógicas e Aritméticas

- Além de operações pixel-a-pixel, também é possível realizar operações orientadas à vizinhança

# Fundamentos de Imagens Digitais

## Operações Lógicas e Aritméticas

- Além de operações pixel-a-pixel, também é possível realizar operações orientadas à vizinhança
- Processo: uso de “máscaras” (ou janelas ou filtros) que são sub-regiões de amostragem para a qual um certo processamento é realizado

# Fundamentos de Imagens Digitais

## Operações Lógicas e Aritméticas

- Além de operações pixel-a-pixel, também é possível realizar operações orientadas à vizinhança
- Processo: uso de “máscaras” (ou janelas ou filtros) que são sub-regiões de amostragem para a qual um certo processamento é realizado
- Exemplo: para uma vizinhança  $3 \times 3$  substitui-se o elemento na posição  $f(x, y)$  pela média de seus 9 vizinhos:

$$\bar{f}(x, y) = \frac{1}{9} \sum_{m=-1}^1 \sum_{n=-1}^1 f(x + m, y + n)$$

# Fundamentos de Imagens Digitais

## Operações Lógicas e Aritméticas

- Além de operações pixel-a-pixel, também é possível realizar operações orientadas à vizinhança
- Processo: uso de “máscaras” (ou janelas ou filtros) que são sub-regiões de amostragem para a qual um certo processamento é realizado
- Exemplo: para uma vizinhança  $3 \times 3$  substitui-se o elemento na posição  $f(x, y)$  pela média de seus 9 vizinhos:

$$\bar{f}(x, y) = \frac{1}{9} \sum_{m=-1}^1 \sum_{n=-1}^1 f(x + m, y + n)$$

$$\begin{aligned}\bar{f}(x, y) = \frac{1}{9} & [f(x - 1, y - 1) + f(x, y - 1) + f(x + 1, y - 1) + f(x - 1, y) + \\ & f(x, y) + f(x + 1, y) + f(x - 1, y + 1) + f(x, y + 1) + f(x + 1, y + 1)]\end{aligned}$$

# Realce de Imagens

# Realce de Imagens

## Realce

Técnicas de realce buscam acentuar ou melhorar a aparência de determinadas características da imagem, tornando-a mais adequada para a aplicação em questão

# Realce de Imagens

## Realce

Técnicas de realce buscam acentuar ou melhorar a aparência de determinadas características da imagem, tornando-a mais adequada para a aplicação em questão

- Quando o realce é necessário ?

# Realce de Imagens

## Realce

Técnicas de realce buscam acentuar ou melhorar a aparência de determinadas características da imagem, tornando-a mais adequada para a aplicação em questão

- Quando o realce é necessário ?
- Sempre que ocorrer degradação ou perda de qualidade devido a:
  - ruído
  - perda de contraste
  - borramento
  - distorção devido ao equipamento de aquisição
  - condições inadequadas de iluminação

# Realce de Imagens

## Realce

Técnicas de realce buscam acentuar ou melhorar a aparência de determinadas características da imagem, tornando-a mais adequada para a aplicação em questão

- Quando o realce é necessário ?
- Sempre que ocorrer degradação ou perda de qualidade devido a:
  - ruído
  - perda de contraste
  - borramento
  - distorção devido ao equipamento de aquisição
  - condições inadequadas de iluminação

Domínio Espacial processamento baseado na manipulação direta dos pixels

# Realce de Imagens

## Realce

Técnicas de realce buscam acentuar ou melhorar a aparência de determinadas características da imagem, tornando-a mais adequada para a aplicação em questão

- Quando o realce é necessário ?
- Sempre que ocorrer degradação ou perda de qualidade devido a:
  - ruído
  - perda de contraste
  - borramento
  - distorção devido ao equipamento de aquisição
  - condições inadequadas de iluminação

Domínio Espacial processamento baseado na manipulação direta dos pixels

Domínio de Frequência processamento baseado na modificação da imagem com a aplicação de transformadas (ex: Fourier)

# Realce de Imagens

## Brilho & Contraste

### Brilho

Sensação visual da intensidade luminosa de uma fonte

# Realce de Imagens

## Brilho & Contraste

### Brilho

Sensação visual da intensidade luminosa de uma fonte

- A habilidade do sistema visual humano em perceber diferentes níveis de brilho deve ser considerada na hora de apresentar resultados que envolvem imagens digitais

# Realce de Imagens

## Brilho & Contraste

### Brilho

Sensação visual da intensidade luminosa de uma fonte

- A habilidade do sistema visual humano em perceber diferentes níveis de brilho deve ser considerada na hora de apresentar resultados que envolvem imagens digitais
- Evidências experimentais: sensibilidade tem resposta logarítmica em relação à intensidade de luz incidente

# Realce de Imagens

## Brilho & Contraste

### Brilho

Sensação visual da intensidade luminosa de uma fonte

- A habilidade do sistema visual humano em perceber diferentes níveis de brilho deve ser considerada na hora de apresentar resultados que envolvem imagens digitais
- Evidências experimentais: sensibilidade tem resposta logarítmica em relação à intensidade de luz incidente
- O fato dessa resposta não ser linear produz uma série de fenômenos ópticos observáveis

# Realce de Imagens

## Brilho & Contraste

### Bandas de Mach

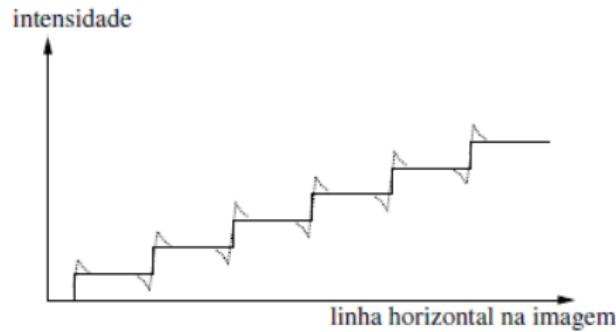
É baseado no princípio de que o sistema visual tende a subestimar ou superestimar a intensidade próxima às transições entre regiões de intensidades diferentes

# Realce de Imagens

## Brilho & Contraste

### Bandas de Mach

É baseado no princípio de que o sistema visual tende a subestimar ou superestimar a intensidade próxima às transições entre regiões de intensidades diferentes



# Realce de Imagens

## Brilho & Contraste

### Contraste Simultâneo

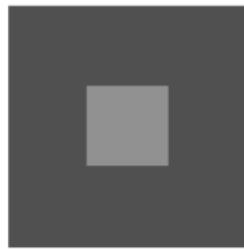
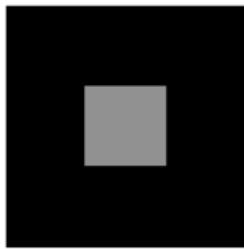
Está relacionado ao fato de que o brilho aparente de uma região depende fortemente da intensidade do fundo

# Realce de Imagens

## Brilho & Contraste

### Contraste Simultâneo

Está relacionado ao fato de que o brilho aparente de uma região depende fortemente da intensidade do fundo



# Realce de Imagens

## Brilho & Contraste

### Benussi-Koffka

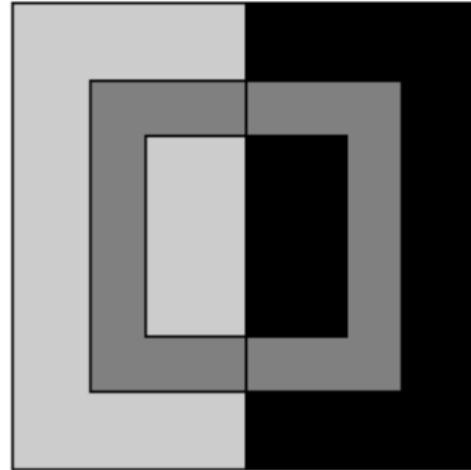
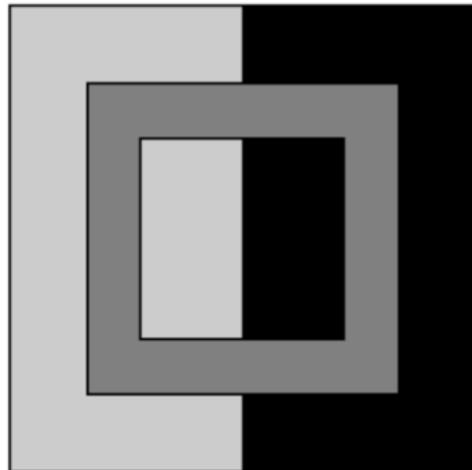
Em que um anel com intensidade uniforme aparenta ter brilho diferente quando uma reta separa os fundos da imagem

# Realce de Imagens

## Brilho & Contraste

### Benussi-Koffka

Em que um anel com intensidade uniforme aparenta ter brilho diferente quando uma reta separa os fundos da imagem



# Realce de Imagens

## Brilho & Contraste

### Contraste

Pode ser definido como uma medida da variação relativa da luminância, ou seja, da intensidade luminosa por unidade de área

# Realce de Imagens

## Brilho & Contraste

### Contraste

Pode ser definido como uma medida da variação relativa da luminância, ou seja, da intensidade luminosa por unidade de área

Lei de Weber a resposta do sistema visual humano depende significativamente de variações locais de luminância, ao invés da luminância absoluta

# Realce de Imagens

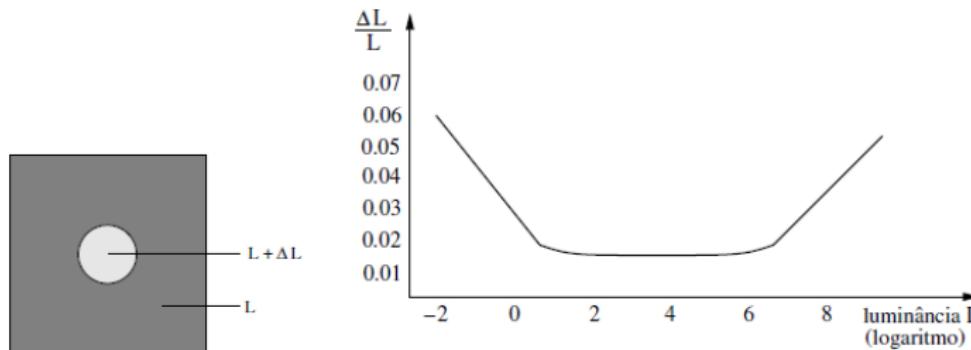
## Brilho & Contraste

- Uma experimento que determina a habilidade do sistema visual humano em discriminar mudanças de luminância, consiste em se apresentar a um observador um objeto com luminância  $L_F = L + \Delta L$  em um fundo  $L_B = L$

# Realce de Imagens

## Brilho & Contraste

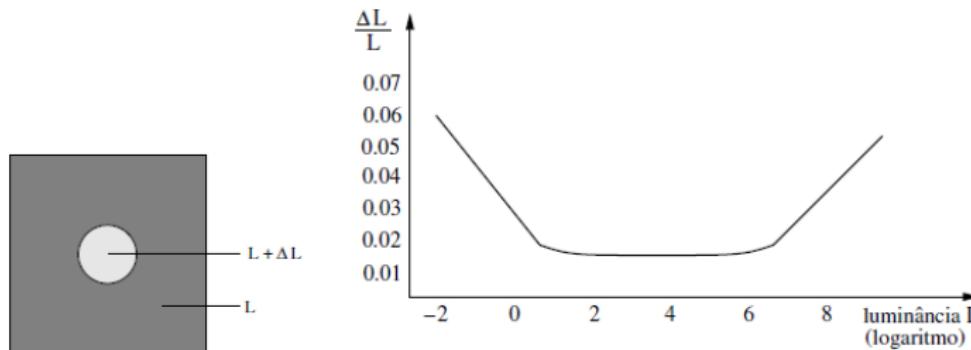
- Uma experiência que determina a habilidade do sistema visual humano em discriminar mudanças de luminância, consiste em se apresentar a um observador um objeto com luminância  $L_F = L + \Delta L$  em um fundo  $L_B = L$



# Realce de Imagens

## Brilho & Contraste

- Uma experiência que determina a habilidade do sistema visual humano em discriminar mudanças de luminância, consiste em se apresentar a um observador um objeto com luminância  $L_F = L + \Delta L$  em um fundo  $L_B = L$



Questão de interesse identificar qual a diferença  $\Delta L = L_F - L_B$  permite o discernimento entre objeto/fundo

# Realce de Imagens

## Brilho & Contraste

- A lei de Weber estabelece que a intensidade adicional de estímulo (luminância) necessária para que o sistema visual humano possa observar uma alteração é proporcional à intensidade inicial, em uma relação constante = **contraste de Weber** ou **Lei Weber-Fechner**

$$C_W = \frac{\Delta L}{L}$$

# Realce de Imagens

## Brilho & Contraste

- A lei de Weber estabelece que a intensidade adicional de estímulo (luminância) necessária para que o sistema visual humano possa observar uma alteração é proporcional à intensidade inicial, em uma relação constante = **contraste de Weber** ou **Lei Weber-Fechner**

$$C_W = \frac{\Delta L}{L}$$

- Devido à capacidade de adaptação do sistema visual humano, a nossa capacidade de percepção do contraste mínimo não se altera em grandes intervalos de intensidades:  $C_W \approx 2\%$

# Realce de Imagens

## Brilho & Contraste

- Outra medida de contraste (**contraste de Michelson**) utilizada em padrões periódicos (ex: senoidais) é dado por

$$C_M = \frac{L_{max} - L_{min}}{L_{max} + L_{min}}$$

onde  $L_{min}$  e  $L_{max}$  correspondem a luminância (intensidade de luz por unidade de área)

# Realce de Imagens

## Brilho & Contraste

- Outra medida de contraste (**contraste de Michelson**) utilizada em padrões periódicos (ex: senoidais) é dado por

$$C_M = \frac{L_{max} - L_{min}}{L_{max} + L_{min}}$$

onde  $L_{min}$  e  $L_{max}$  correspondem a luminância (intensidade de luz por unidade de área)

- $C_W$  e  $C_M$  não são equivalentes

# Realce de Imagens

## Brilho & Contraste

- Outra medida de contraste (**contraste de Michelson**) utilizada em padrões periódicos (ex: senoidais) é dado por

$$C_M = \frac{L_{max} - L_{min}}{L_{max} + L_{min}}$$

onde  $L_{min}$  e  $L_{max}$  correspondem a luminância (intensidade de luz por unidade de área)

- $C_W$  e  $C_M$  não são equivalentes
- $C_M$  varia no intervalo  $[0 \dots 1]$ , enquanto que  $C_W$  varia no intervalo  $[-1 \dots \infty)$

# Realce de Imagens

## Brilho & Contraste

- Outra medida de contraste (**contraste de Michelson**) utilizada em padrões periódicos (ex: senoidais) é dado por

$$C_M = \frac{L_{max} - L_{min}}{L_{max} + L_{min}}$$

onde  $L_{min}$  e  $L_{max}$  correspondem a luminância (intensidade de luz por unidade de área)

- $C_W$  e  $C_M$  não são equivalentes
- $C_M$  varia no intervalo  $[0 \dots 1]$ , enquanto que  $C_W$  varia no intervalo  $[-1 \dots \infty)$
- Essas medidas são boas para padrões simples mas falham como preditores de contrastes em padrões mais complexos

# Realce de Imagens

## Brilho & Contraste

- Medida de **contraste local de Beghdadi e Khellaf** dado um pixel  $f$  com coordenadas  $(x, y)$ , centrado na vizinhança  $w_f$  é dada por:

$$C_f = \frac{|l_{(x,y)} - \bar{l}_{(x,y)}|}{\bar{l}_{(x,y)}}$$

onde  $l_{(x,y)}$  é o nível de cinza do pixel e  $\bar{l}_{(x,y)}$  é o nível de cinza médio na vizinhança  $w_f$

# Realce de Imagens

## Histograma

### Histograma

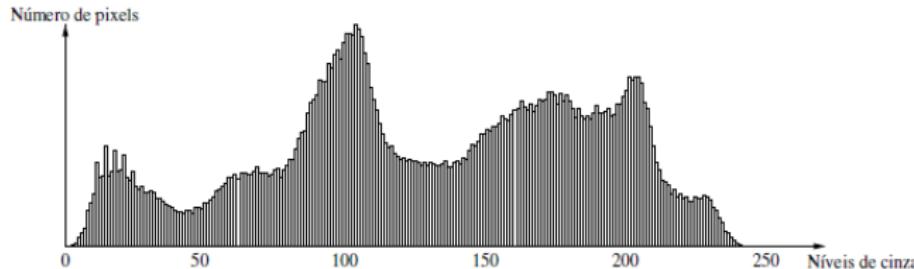
Corresponde à distribuição de frequência dos níveis de cinza de uma imagem

# Realce de Imagens

## Histograma

### Histograma

Corresponde à distribuição de frequência dos níveis de cinza de uma imagem



# Realce de Imagens

## Histograma

- Seja uma imagem representada por uma matriz bidimensional, denotada por  $f(x, y)$  de tamanho  $M \times N$  e contendo  $L$  níveis de cinza no intervalo  $[0 \dots L_{max}]$

# Realce de Imagens

## Histograma

- Seja uma imagem representada por uma matriz bidimensional, denotada por  $f(x, y)$  de tamanho  $M \times N$  e contendo  $L$  níveis de cinza no intervalo  $[0 \dots L_{max}]$
- O cálculo do histograma dessa imagem é obtido a partir do seguinte algoritmo:

```
1      for i in range(Lmax):
2          H[i] = 0
3
4      for x in range(M-1):
5          for y in range(N-1):
6              H[f(x,y)] = H[f(x,y)] + 1
7
```

# Realce de Imagens

## Histograma

- Uma imagem possui um único histograma porém o contrário não é verdade

# Realce de Imagens

## Histograma

- Uma imagem possui um único histograma porém o contrário não é verdade
- Um histograma é uma distribuição discreta de probabilidade = define a chance de se encontrar um pixel com um determinado nível de cinza em uma imagem  $f$ :

$$p_k(f) = \frac{H(k)}{M \times N}$$

# Realce de Imagens

## Histograma

- Uma imagem possui um único histograma porém o contrário não é verdade
- Um histograma é uma distribuição discreta de probabilidade = define a chance de se encontrar um pixel com um determinado nível de cinza em uma imagem  $f$ :

$$p_k(f) = \frac{H(k)}{M \times N}$$

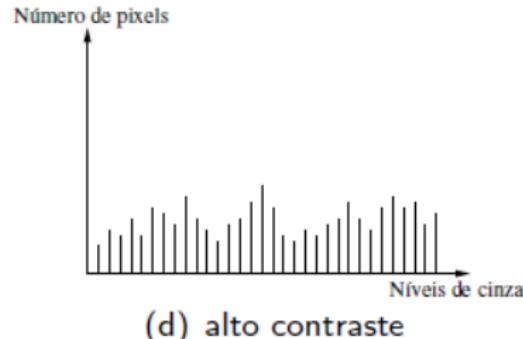
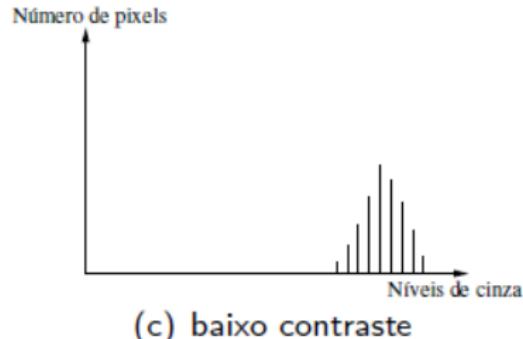
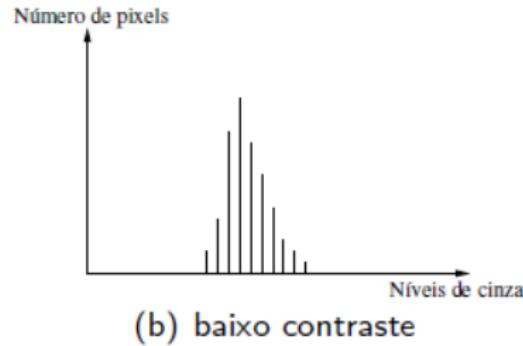
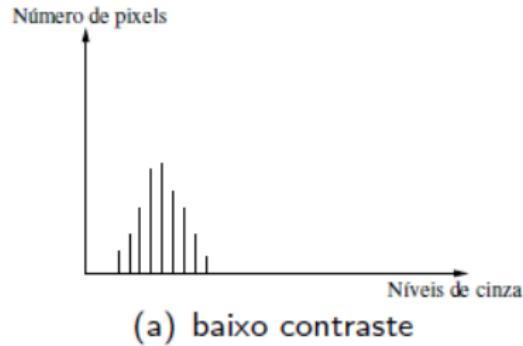
$H(k)$  quantidade de pixels na imagem com nível de cinza  $k$

$M$  resolução horizontal da imagem

$N$  resolução vertical da imagem

# Realce de Imagens

## Histograma vs Contraste



# Realce de Imagens

## Histograma - Exercício

- Usando Numpy, extrair as informações de histograma de uma imagem:

# Realce de Imagens

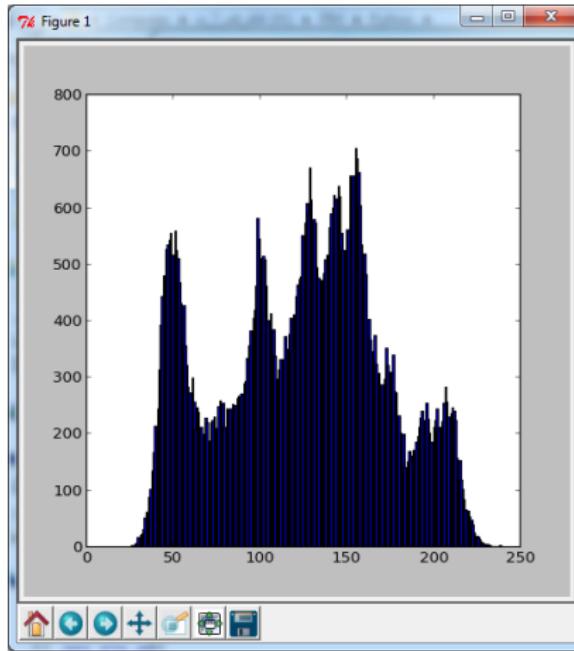
## Histograma - Exercício

- Usando Numpy, extrair as informações de histograma de uma imagem:

```
1 import numpy as np
2 import scipy.misc
3 import matplotlib.pyplot as plt
4
5 path = "imagens/"
6 imagem = scipy.misc.imread(path+"lena.jpg")
7
8 # gera o histograma e seus intervalos
9 histograma, intervalos = np.histogram(imagem,bins=np.arange(0,256))
10
11 # determina o valor do centro dos intervalos
12 center = (intervalos[:-1] + intervalos[1:])/2
13
14 # exibe o histograma
15 plt.bar(center, histograma, align='center')
16 plt.show()
```

# Realce de Imagens

## Histograma - Exercício



# Realce de Imagens

## Transformação da Escala de Cinza

- O objetivo das transformações de contraste é melhorar a qualidade da imagem = enfatizar informações de interesse

# Realce de Imagens

## Transformação da Escala de Cinza

- O objetivo das transformações de contraste é melhorar a qualidade da imagem = enfatizar informações de interesse
- **Contraste:** diferença entre os valores máximo e mínimo nos níveis de cinza em  $f(x, y)$

# Realce de Imagens

## Transformação da Escala de Cinza

- O objetivo das transformações de contraste é melhorar a qualidade da imagem = enfatizar informações de interesse
- **Contraste:** diferença entre os valores máximo e mínimo nos níveis de cinza em  $f(x, y)$
- Se a imagem não ocupar todo o espectro de níveis de cinza disponível pode-se realizar uma operação de *ampliação de contraste* através do mapeamento do intervalo de níveis de cinza  $[L_{min}, L_{max}]$  da imagem original para o intervalo disponível que geralmente é  $[0, 255]$

# Realce de Imagens

## Transformação da Escala de Cinza

- O objetivo das transformações de contraste é melhorar a qualidade da imagem = enfatizar informações de interesse
- **Contraste:** diferença entre os valores máximo e mínimo nos níveis de cinza em  $f(x, y)$
- Se a imagem não ocupar todo o espectro de níveis de cinza disponível pode-se realizar uma operação de *ampliação de contraste* através do mapeamento do intervalo de níveis de cinza  $[L_{min}, L_{max}]$  da imagem original para o intervalo disponível que geralmente é  $[0, 255]$

# Realce de Imagens

## Transformação da Escala de Cinza

- Uma função de transformação de contraste  $T$  pode ser descrita na forma  $g = T(f)$ , onde:

# Realce de Imagens

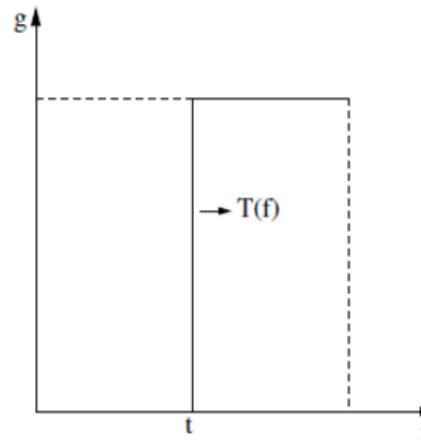
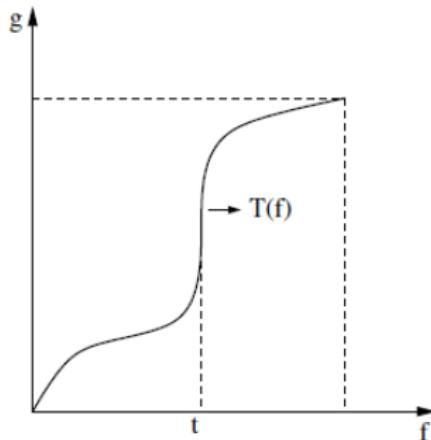
## Transformação da Escala de Cinza

- Uma função de transformação de contraste  $T$  pode ser descrita na forma  $g = T(f)$ , onde:
  - $f$  representa o nível de cinza na imagem original
  - $g$  representa o novo nível de cinza transformado

# Realce de Imagens

## Transformação da Escala de Cinza

- Uma função de transformação de contraste  $T$  pode ser descrita na forma  $g = T(f)$ , onde:
  - $f$  representa o nível de cinza na imagem original
  - $g$  representa o novo nível de cinza transformado
- Exemplos:



# Realce de Imagens

## Transformação da Escala de Cinza (1/2)

```
1 import numpy as np
2 import scipy .misc
3 import matplotlib .pyplot as plt
4
5 def Binariza( imagem, threshold ):
6     shape = imagem.shape
7
8     imagemBin = []
9     for L in range( shape[0] ):
10        row = []
11        for C in range( shape[1] ):
12            if( imagem[L][C] <= threshold ):
13                row.append( [0, 0, 0] )
14            else:
15                row.append( [255, 255, 255] )
16            imagemBin.append( row )
17
return imagemBin
```

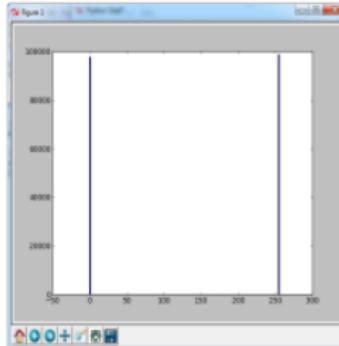
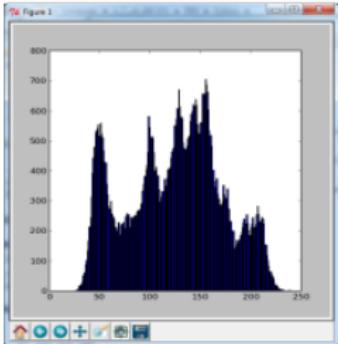
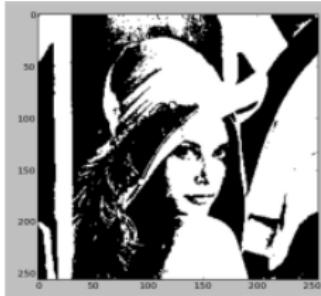
# Realce de Imagens

## Transformação da Escala de Cinza (2/2)

```
21 imagem = scipy.misc.imread( "imagens/lena.jpg" )  
22 scipy .misc.imsave( "imagens/lena-binaria.png", Binariza( imagem, 128 ) )  
23  
24 plt.imshow( scipy .misc.imread( "imagens/lena-binaria.png" ) )  
25 plt.show()
```

# Realce de Imagens

Transformação da Escala de Cinza



# Realce de Imagens

## Transformações Lineares e Não-Lineares

- Uma função de transformação linear pode ser descrita na forma  $g = af + b$ , onde:

# Realce de Imagens

## Transformações Lineares e Não-Lineares

- Uma função de transformação linear pode ser descrita na forma  $g = af + b$ , onde:
  - **a** controla a escala de níveis de cinza da imagem
  - **b** ajuste de brilho da imagem

# Realce de Imagens

## Transformações Lineares e Não-Lineares

- Uma função de transformação linear pode ser descrita na forma  $g = af + b$ , onde:
  - **a** controla a escala de níveis de cinza da imagem
  - **b** ajuste de brilho da imagem
- Seja uma imagem  $f$  onde seus níveis de cinza variam entre  $[f_{min}, f_{max}]$  que se deseja mapear sobre uma nova imagem no intervalo  $[g_{min}, g_{max}]$  temos:

$$g = \frac{g_{max} - g_{min}}{f_{max} - f_{min}}(f_{xy} - f_{min}) + g_{min}$$

# Realce de Imagens

## Transformações Lineares e Não-Lineares

- Uma função de transformação linear pode ser descrita na forma  $g = af + b$ , onde:
  - **a** controla a escala de níveis de cinza da imagem
  - **b** ajuste de brilho da imagem
- Seja uma imagem  $f$  onde seus níveis de cinza variam entre  $[f_{min}, f_{max}]$  que se deseja mapear sobre uma nova imagem no intervalo  $[g_{min}, g_{max}]$  temos:

$$g = \frac{g_{max} - g_{min}}{f_{max} - f_{min}}(f_{xy} - f_{min}) + g_{min}$$

- Considerando  $a = \frac{g_{max} - g_{min}}{f_{max} - f_{min}}$  então pode-se perceber que:

$$\left\{ \begin{array}{ll} a = 1 & \text{O intervalo não se altera, apenas se desloca} \\ a < 1 & \text{O intervalo é comprimido em seu tamanho} \\ a > 1 & \text{O intervalo é expandido em seu tamanho} \end{array} \right.$$

# Realce de Imagens

## Transformações Lineares e Não-Lineares

- Uma **transformação linear por partes** é caracterizada pela composição de múltiplas transformações lineares

# Realce de Imagens

## Transformações Lineares e Não-Lineares

- Uma **transformação linear por partes** é caracterizada pela composição de múltiplas transformações lineares
- Realce baseado em intervalos específicos de níveis de cinza

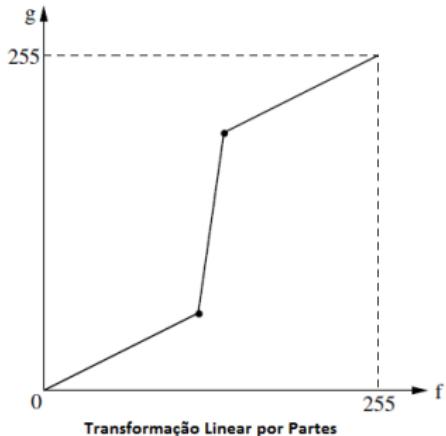
# Realce de Imagens

## Transformações Lineares e Não-Lineares

- Uma **transformação linear por partes** é caracterizada pela composição de múltiplas transformações lineares
- Realce baseado em intervalos específicos de níveis de cinza
- Permitir realçar aspectos específicos na imagem ou corrigir irregularidades muito acentuadas na imagem

# Realce de Imagens

Transformações Lineares e Não-Lineares



# Realce de Imagens

## Transformações Lineares e Não-Lineares

- A **transformação linear inversa** é caracterizada pela reflexão horizontal do histograma da imagem

# Realce de Imagens

## Transformações Lineares e Não-Lineares

- A **transformação linear inversa** é caracterizada pela reflexão horizontal do histograma da imagem
- Níveis claros se tornam escuros e vice-versa → negativo da imagem

# Realce de Imagens

## Transformações Lineares e Não-Lineares

- A **transformação linear inversa** é caracterizada pela reflexão horizontal do histograma da imagem
- Níveis claros se tornam escuros e vice-versa → negativo da imagem
- Dado por:

$$g = (f_{max} - f) + f_{min}$$

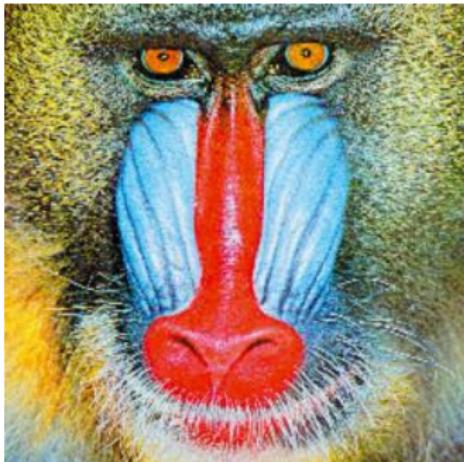
# Realce de Imagens

## Transformações Lineares Inversa: Exercício

```
1 import numpy as np
2 import scipy .misc
3 import matplotlib .pyplot as plt
4
5 path = "imagens/"
6 imagem = scipy.misc.imread(path+"mandril.jpg")
7 shape = imagem.shape
8
9 for L in range(shape[0]):
10     for C in range(shape[1]):
11         r,g,b = imagem[L][C]
12         imagem[L][C] = [255-r,255-g,255-b]
13
14 scipy .misc.imsave(path+"mandril-neg.jpg", imagem)
15
16 plt .imshow(imagem)
17 plt .show()
```

# Realce de Imagens

Transformações Lineares Inversa: Exercício



# Realce de Imagens

## Transformações Não-Lineares

- Em uma transformação não-linear, o parâmetro **a** em  $g = af + b$  deixa de ser um valor constante (como ocorre com as transformações lineares)

# Realce de Imagens

## Transformações Não-Lineares

- Em uma transformação não-linear, o parâmetro **a** em  $g = af + b$  deixa de ser um valor constante (como ocorre com as transformações lineares)
  
- Tipos:
  - Logaritmo
  - Raiz Quadrada
  - Exponencial
  - Quadrado

# Realce de Imagens

## Transformações Não-Lineares: Logaritmo

- Substitui cada valor de pixel na imagem pelo seu logaritmo

# Realce de Imagens

## Transformações Não-Lineares: Logaritmo

- Substitui cada valor de pixel na imagem pelo seu logaritmo
- Propicia um maior realce nos pixels com baixa intensidade, ou seja, nas regiões escuras

# Realce de Imagens

## Transformações Não-Lineares: Logaritmo

- Substitui cada valor de pixel na imagem pelo seu logaritmo
- Propicia um maior realce nos pixels com baixa intensidade, ou seja, nas regiões escuras
- Abordagem usual: usar logaritmo base 10 → a base não influencia na curva em si, apenas na escala de tons

# Realce de Imagens

## Transformações Não-Lineares: Logaritmo

- Substitui cada valor de pixel na imagem pelo seu logaritmo
- Propicia um maior realce nos pixels com baixa intensidade, ou seja, nas regiões escuras
- Abordagem usual: usar logaritmo base 10 → a base não influencia na curva em si, apenas na escala de tons
- E como  $\log 0 = \text{?}$  usa-se um fator de correção ( $a$ ) na fórmula:  
$$g = a \log(f + 1)$$
 para garantir que a função esteja no intervalo  $[0, 255]$

# Realce de Imagens

## Transformações Não-Lineares: Logaritmo

- Substitui cada valor de pixel na imagem pelo seu logaritmo
- Propicia um maior realce nos pixels com baixa intensidade, ou seja, nas regiões escuras
- Abordagem usual: usar logaritmo base 10 → a base não influencia na curva em si, apenas na escala de tons
- E como  $\log 0 = \frac{1}{0}$  usa-se um fator de correção ( $a$ ) na fórmula:  
$$g = a \log(f + 1)$$
 para garantir que a função esteja no intervalo  $[0, 255]$
- Se  $f_{max}$  for a maior intensidade de cor então  $a = \frac{255}{\log(f_{max}+1)}$

# Realce de Imagens

## Transformações Não-Lineares: Exponencial

- Substitui cada valor de pixel na imagem pelo seu exponencial

# Realce de Imagens

## Transformações Não-Lineares: Exponencial

- Substitui cada valor de pixel na imagem pelo seu exponencial
- Propicia um maior realce nos pixels com alta intensidade, ou seja, nas regiões claras

# Realce de Imagens

## Transformações Não-Lineares: Exponencial

- Substitui cada valor de pixel na imagem pelo seu exponencial
- Propicia um maior realce nos pixels com alta intensidade, ou seja, nas regiões claras
- A fórmula:  $g = T(f) = a(e^f - 1)$

# Realce de Imagens

## Transformações Não-Lineares: Quadrado

- É dada pela fórmula:  $g = T(f) = af^2$

# Realce de Imagens

## Transformações Não-Lineares: Quadrado

- É dada pela fórmula:  $g = T(f) = af^2$
- Semelhante à transformação exponencial: aumenta o contraste entre as regiões de média e alta intensidades

# Realce de Imagens

## Transformações Não-Lineares: Raiz Quadrada

- É dada pela fórmula:  $g = T(f) = a\sqrt{f}$

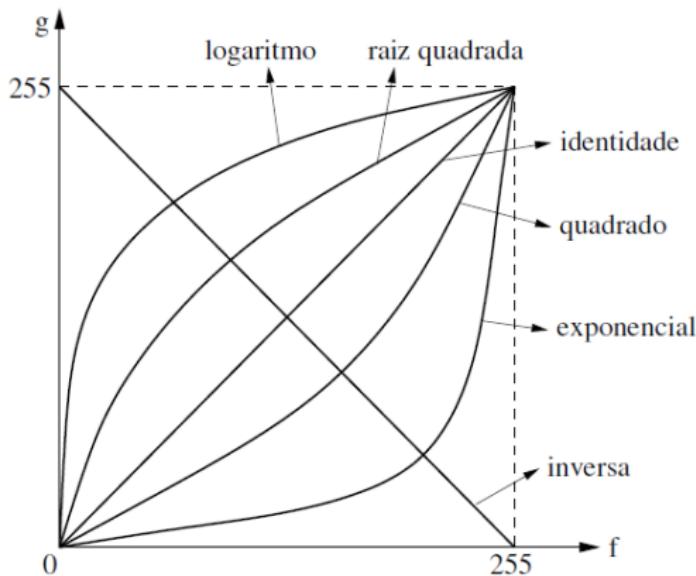
# Realce de Imagens

## Transformações Não-Lineares: Raiz Quadrada

- É dada pela fórmula:  $g = T(f) = a\sqrt{f}$
- Semelhante à transformação logaritmo: aumenta o contraste entre as regiões de média e baixa intensidades

# Realce de Imagens

## Transformações Não-Lineares



# Realce de Imagens

## Transformações Não-Lineares



Imagen Original



Logaritmo



Raiz Quadrada



Exponencial



Quadrado

# Realce de Imagens

## Transformações Não-Lineares: Exemplo Prático

```
1 import numpy as np
2 import scipy.misc
3
4 original = scipy.misc.imread( "imagens/lena.jpg" )
5 a = 255.0/np.log(256) # fmax + 1 = 255 + 1
6 transformada = np.log( original ) * a
7 scipy.misc.imsave("imagens/lena-log.jpg", transformada)
```

## Transformações Não-Lineares: Logaritmo

# Realce de Imagens

Transformações Não-Lineares



# Realce de Imagens

## Equalização do Histograma

### Equalização do Histograma

Modifica o histograma da imagem  $f$  de maneira que a imagem resultante  $g$  possua uma distribuição mais uniforme dos seus níveis de cinza  $\equiv$  todos os valores são aproximados com a mesma frequência

# Realce de Imagens

## Equalização do Histograma

- Dado que os níveis de cinza normalizados de uma imagem são valores aleatórios entre  $[0, 1]$ , temos então as funções densidade de probabilidade  $p_f(f)$  e  $p_g(g)$

# Realce de Imagens

## Equalização do Histograma

- Dado que os níveis de cinza normalizados de uma imagem são valores aleatórios entre  $[0, 1]$ , temos então as funções densidade de probabilidade  $p_f(f)$  e  $p_g(g)$

$$p_g(g) = \left[ p_f(f) \frac{df}{dg} \right]_{f=T^{-1}(g)} \quad (1)$$

onde  $f = T^{-1}(g)$  é a transformação inversa dos níveis de cinza de  $g$  para  $f$

# Realce de Imagens

## Equalização do Histograma

- Dado que os níveis de cinza normalizados de uma imagem são valores aleatórios entre  $[0, 1]$ , temos então as funções densidade de probabilidade  $p_f(f)$  e  $p_g(g)$

$$p_g(g) = \left[ p_f(f) \frac{df}{dg} \right]_{f=T^{-1}(g)} \quad (1)$$

onde  $f = T^{-1}(g)$  é a transformação inversa dos níveis de cinza de  $g$  para  $f$

- Exemplo de transformação  $T$  (*Função de Distribuição Acumulada*):

$$g = T(f) = \int_0^f p_f(w)dw \quad 0 \leq f \leq 1 \quad (2)$$

onde  $w$  é a variável de integração

# Realce de Imagens

## Equalização do Histograma

- Para que seja útil ao processamento de imagens, é necessário que tratemos essas equações em sua forma discreta:

# Realce de Imagens

## Equalização do Histograma

- Para que seja útil ao processamento de imagens, é necessário que tratemos essas equações em sua forma discreta:

$$g_k = \sum_{i=0}^k p_f(f_i) = \sum_{i=0}^k \frac{n_i}{n} \quad k = 0, 1, \dots, L - 1 \quad (3)$$

onde:  $p_f(f_i)$  é a probabilidade do i-ésimo nível de cinza;  $n_i$  é o número de ocorrências do nível de cinza  $i$ ,  $n$  é a quantidade de pixels da imagem (dado por  $n = M \times N$ ) e  $k$  indica um dado nível de cinza com valores entre 0 e  $L - 1$

# Realce de Imagens

## Equalização do Histograma: Algoritmo

- ➊ Calcular o histograma da imagem a ser transformada
- ➋ Normalizar o histograma de maneira que cada nível de cinza esteja entre  $0 \leq f_k \leq 1$
- ➌ for k in range(0, L-1):
  - ➍ Calcular a função de Distribuição Acumulada de Probabilidade

$$g_k = \sum_{i=0}^k p_f(f_i)$$

- ➎ Arredondar os valores para o nível de cinza mais próximo

$$g_k = \text{round}(g_k \times L_{\max})$$

- ➏ Agrupar valores  $f_k$  para formar  $g_k$

# Realce de Imagens

## Equalização do Histograma: Exemplo Prático

- Considere uma imagem com 8 níveis de cinza que apresente o seguinte histograma abaixo:

Histograma a ser equilizado

Níveis de cinza ( $k$ )	0	1	2	3	4	5	6	7
# de Pixels ( $n_k$ )	1314	3837	5820	4110	2374	921	629	516

# Realce de Imagens

## Equalização do Histograma: Exemplo Prático

Passo #1 - Encontrar as probabilidades  $p_f$  para cada nível de cinza  $k$

$$p_f(f_0) = 1314/19521 \approx 0.067$$

$$p_f(f_1) = 3837/19521 \approx 0.197$$

$$p_f(f_2) = 5820/19521 \approx 0.298$$

$$p_f(f_3) = 4110/19521 \approx 0.211$$

$$p_f(f_4) = 2374/19521 \approx 0.122$$

$$p_f(f_5) = 921/19521 \approx 0.047$$

$$p_f(f_6) = 629/19521 \approx 0.032$$

$$p_f(f_7) = 516/19521 \approx 0.026$$

# Realce de Imagens

## Equalização do Histograma: Exemplo Prático

Passo #2 - Calcular as funções de distribuição acumulada de probabilidade

$$g_0 = T(f_0) = \sum_{i=0}^0 p_f(f_0) = 0.067$$

$$g_1 = T(f_1) = \sum_{i=0}^1 p_f(f_1) = 0.264$$

⋮

# Realce de Imagens

## Equalização do Histograma: Exemplo Prático

Passo #2 - Calcular as funções de distribuição acumulada de probabilidade

$$g_0 = T(f_0) = \sum_{i=0}^0 p_f(f_0) = 0.067$$

$$g_1 = T(f_1) = \sum_{i=0}^1 p_f(f_1) = 0.264$$

⋮

De forma similar, temos:

$$g_2 = 0.562 \quad g_3 = 0.773 \quad g_4 = 0.895$$

$$g_5 = 0.942 \quad g_6 = 0.974 \quad g_7 = 1.000$$

# Realce de Imagens

## Equalização do Histograma: Exemplo Prático

### Passo #3 - Arredondar para o nível de cinza mais próximo

$$g_0 = \text{round}(0.067 \times 7) = 0.469 \approx 0$$

$$g_1 = \text{round}(0.264 \times 7) = 1.848 \approx 2$$

$$g_2 = \text{round}(0.562 \times 7) = 3.934 \approx 4$$

$$g_3 = \text{round}(0.773 \times 7) = 5.411 \approx 5$$

$$g_4 = \text{round}(0.895 \times 7) = 6.265 \approx 6$$

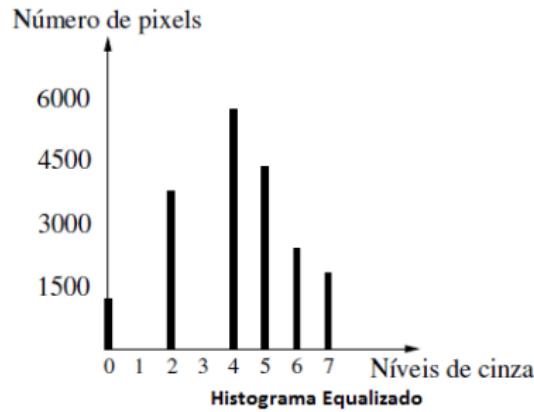
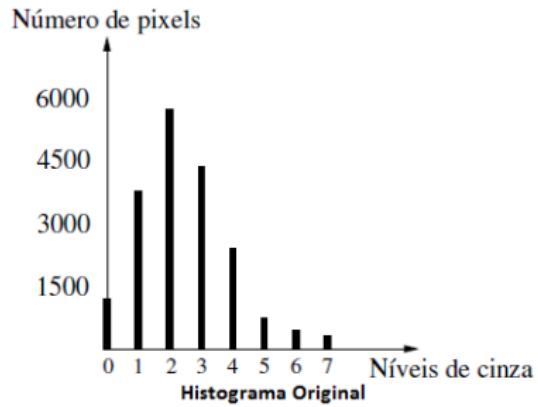
$$g_5 = \text{round}(0.942 \times 7) = 6.594 \approx 7$$

$$g_6 = \text{round}(0.974 \times 7) = 6.818 \approx 7$$

$$g_7 = \text{round}(1.000 \times 7) = 7.000 = 7$$

# Realce de Imagens

Equalização do Histograma: Exemplo Prático



# Realce de Imagens

## Equalização do Histograma: Implementação (1/2)

```
1 import numpy as np
2 import scipy.misc
3
4 car = scipy.misc.imread("imagens/car.png")
5 histograma, intervalos = np.histogram(car, bins=np.arange(0,256))
6
7 # passo 1: calculo das probabilidades
8 npixels = car.shape[0] * car.shape[1] * 1.0 # usado para converter para float
9 probabilidades = histograma / npixels
10
11 FDP = np.zeros(256)
12 for i in range(256):
13     for j in range(i):      # passo 2: calculo da FDP
14         FDP[i] = FDP[i] + probabilidades[j]
15     FDP[i] = round(FDP[i]*255) # passo 3: arredondamento
```

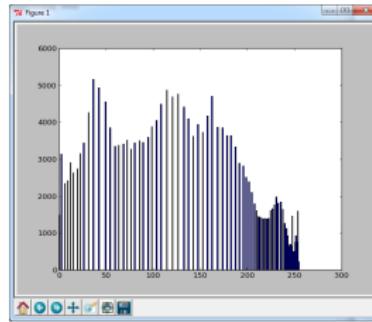
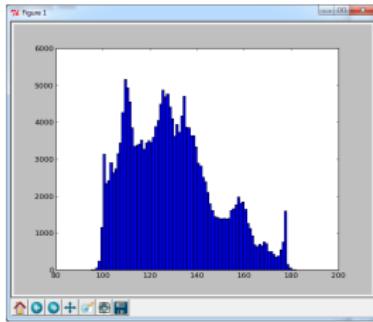
# Realce de Imagens

## Equalização do Histograma: Implementação (2/2)

```
16  # passo 4: agrupar os níveis de cinza
17  car1 = np.zeros(car.shape)
18  for i in range(car.shape[0]):
19      for j in range(car.shape[1]):
20          car1[i][j] = int(FDP[car[i][j]])
21
22  scipy.misc.imsave("imagens/car1.png",car1)
```

# Realce de Imagens

Equalização do Histograma: Resultado



# Realce de Imagens

## Filtragem de Imagens

### Filtros de Imagens

São transformações aplicadas a imagens no domínio espacial, que não dependem apenas do valor do nível de cinza em cada pixel. Os filtros consideram a sua vizinhança para o cálculo.

# Realce de Imagens

## Filtragem de Imagens

### Filtros de Imagens

São transformações aplicadas a imagens no domínio espacial, que não dependem apenas do valor do nível de cinza em cada pixel. Os filtros consideram a sua vizinhança para o cálculo.

Tipos de filtros:

Passa-baixa atenua as altas frequências que estão relacionadas com a informação de detalhes da imagem

# Realce de Imagens

## Filtragem de Imagens

### Filtros de Imagens

São transformações aplicadas a imagens no domínio espacial, que não dependem apenas do valor do nível de cinza em cada pixel. Os filtros consideram a sua vizinhança para o cálculo.

Tipos de filtros:

Passa-baixa atenua as altas frequências que estão relacionadas com a informação de detalhes da imagem

Passa-alta realça as altas frequências e são normalmente usados para realçar os detalhes na imagem

# Realce de Imagens

## Filtragem de Imagens

### Filtros de Imagens

São transformações aplicadas a imagens no domínio espacial, que não dependem apenas do valor do nível de cinza em cada pixel. Os filtros consideram a sua vizinhança para o cálculo.

Tipos de filtros:

Passa-baixa atenua as altas frequências que estão relacionadas com a informação de detalhes da imagem

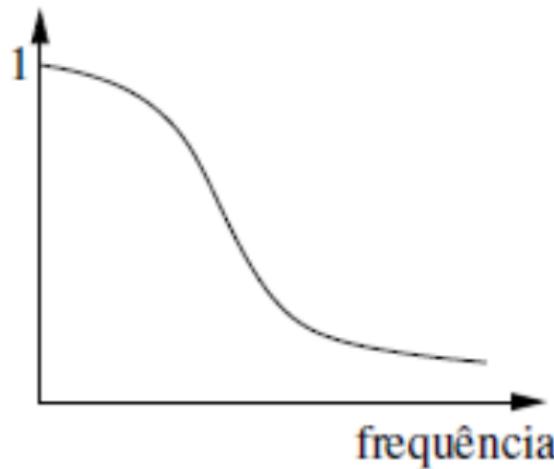
Passa-alta realça as altas frequências e são normalmente usados para realçar os detalhes na imagem

Passa-faixa seleciona um intervalo de frequências do sinal para ser realçado

# Realce de Imagens

## Filtragem de Imagens: filtros passa-baixa

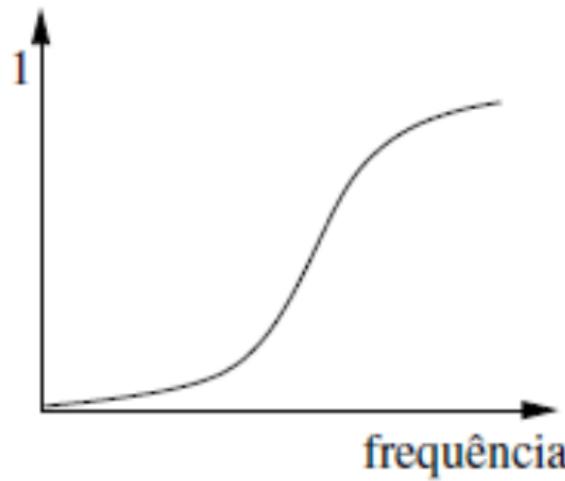
- O efeito visual de um filtro passa-baixas é o de suavização da imagem, uma vez que as altas frequências, que correspondem às transições abruptas, são atenuadas. A suavização tende também, pelas mesmas razões, a minimizar o efeito do ruído em imagens



# Realce de Imagens

Filtragem de Imagens: filtros passa-alta

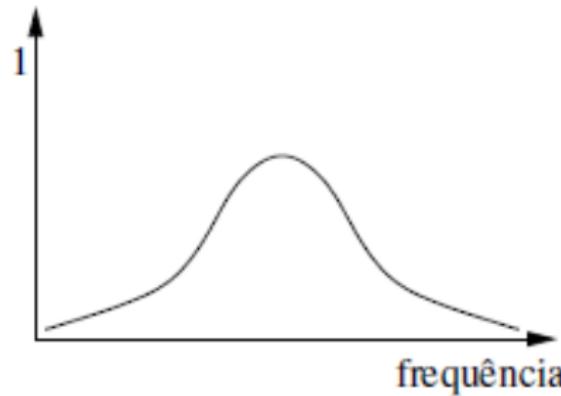
- O efeito obtido é, em geral, o de tornar mais nítidas as transições entre regiões diferentes, conhecidas como bordas. Um efeito indesejado desses filtros é o de enfatizar o ruído presente na imagem



# Realce de Imagens

Filtragem de Imagens: filtros passa-faixa

- O efeito obtido é, em geral, é a ênfase de uma faixa de valores de interesse



# Realce de Imagens

## Filtragem no Domínio Espacial

Domínio Espacial refere-se ao próprio plano da imagem, ou seja, ao conjunto de pixels que compõe uma imagem.

# Realce de Imagens

## Filtragem no Domínio Espacial

Domínio Espacial refere-se ao próprio plano da imagem, ou seja, ao conjunto de pixels que compõe uma imagem.

- O valor do nível de cinza em um ponto  $f(x, y)$  é o resultado da computação de sua vizinhança e da distância dos vizinhos ao ponto de interesse = pontos próximos influenciam mais que os distantes

# Realce de Imagens

## Filtragem no Domínio Espacial

Domínio Espacial refere-se ao próprio plano da imagem, ou seja, ao conjunto de pixels que compõe uma imagem.

- O valor do nível de cinza em um ponto  $f(x, y)$  é o resultado da computação de sua vizinhança e da distância dos vizinhos ao ponto de interesse = pontos próximos influenciam mais que os distantes
- Dois tipos de filtros no domínio espacial: *lineares* e *não-lineares*

# Realce de Imagens

## Filtragem no Domínio Espacial

Domínio Espacial refere-se ao próprio plano da imagem, ou seja, ao conjunto de pixels que compõe uma imagem.

- O valor do nível de cinza em um ponto  $f(x, y)$  é o resultado da computação de sua vizinhança e da distância dos vizinhos ao ponto de interesse = pontos próximos influenciam mais que os distantes
- Dois tipos de filtros no domínio espacial: *lineares* e *não-lineares*
- A matriz que influencia os valores de um pixel no filtro é denominado **máscara**. Cada posição na máscara é denominado *peso* ou *coeficiente*

# Realce de Imagens

## Filtragem no Domínio Espacial

Domínio Espacial refere-se ao próprio plano da imagem, ou seja, ao conjunto de pixels que compõe uma imagem.

- O valor do nível de cinza em um ponto  $f(x, y)$  é o resultado da computação de sua vizinhança e da distância dos vizinhos ao ponto de interesse = pontos próximos influenciam mais que os distantes
- Dois tipos de filtros no domínio espacial: *lineares* e *não-lineares*
- A matriz que influencia os valores de um pixel no filtro é denominado **máscara**. Cada posição na máscara é denominado *peso* ou *coeficiente*
- O nível de cinza de um pixel é o resultado da média ponderada dos vizinhos em relação aos pesos na máscara

# Realce de Imagens

## Filtragem no Domínio Espacial: máscara

- Por exemplo: uma máscara  $3 \times 3$  é denotada por  
 $z_i = f(x, y), 1 \leq i \leq 9$

# Realce de Imagens

## Filtragem no Domínio Espacial: máscara

- Por exemplo: uma máscara  $3 \times 3$  é denotada por  
$$z_i = f(x, y), 1 \leq i \leq 9$$

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum_{i=1}^9 w_i z_i$$

onde  $w_i$  representa os coeficientes da máscara

# Realce de Imagens

## Filtragem no Domínio Espacial: máscara

- Por exemplo: uma máscara  $3 \times 3$  é denotada por  
$$z_i = f(x, y), 1 \leq i \leq 9$$

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum_{i=1}^9 w_i z_i$$

onde  $w_i$  representa os coeficientes da máscara

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

# Realce de Imagens

## Filtragem no Domínio Espacial: máscara

- Por exemplo: uma máscara  $3 \times 3$  é denotada por  

$$z_i = f(x, y), 1 \leq i \leq 9$$

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum_{i=1}^9 w_i z_i$$

onde  $w_i$  representa os coeficientes da máscara

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

- O centro da máscara é posicionado em um pixel  $P$  da imagem; a seguir, o valor de  $P$  é substituído por  $R$ . Finalmente, a máscara é deslocada para o próximo pixel

# Realce de Imagens

## Correlação e Convolução

- A correlação mais é a *filtragem de média* que, no caso unidimensional temos que o valor do nível de cinza de um pixel é obtida pela média de seus dois vizinhos

# Realce de Imagens

## Correlação e Convolução

- A correlação mais é a *filtragem de média* que, no caso unidimensional temos que o valor do nível de cinza de um pixel é obtida pela média de seus dois vizinhos
- Seja a imagem unidimensional  $f$  representada pelo vetor:

...	5	4	2	3	7	4	6	5	4	6	...
-----	---	---	---	---	---	---	---	---	---	---	-----

# Realce de Imagens

## Correlação e Convolução

- A correlação mais é a *filtragem de média* que, no caso unidimensional temos que o valor do nível de cinza de um pixel é obtida pela média de seus dois vizinhos
- Seja a imagem unidimensional  $f$  representada pelo vetor:

...	5	4	2	3	7	4	6	5	4	6	...
-----	---	---	---	---	---	---	---	---	---	---	-----

- O cálculo da média para o pixel de valor 3 produzirá o valor 4 que é a média entre 2, 3, e 7. Neste caso considera-se apenas uma janela de tamanho 3

# Realce de Imagens

## Correlação e Convolução

- A correlação mais é a *filtragem de média* que, no caso unidimensional temos que o valor do nível de cinza de um pixel é obtida pela média de seus dois vizinhos
- Seja a imagem unidimensional  $f$  representada pelo vetor:

...	5	4	2	3	7	4	6	5	4	6	...
-----	---	---	---	---	---	---	---	---	---	---	-----

- O cálculo da média para o pixel de valor 3 produzirá o valor 4 que é a média entre 2, 3, e 7. Neste caso considera-se apenas uma janela de tamanho 3
- A filtragem é feita deslocando-se uma máscara de tamanho 3 ( $w = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ ) pela imagem. Em geral, filtros sempre tem tamanho ímpar para produzir um elemento central

# Realce de Imagens

## Correlação e Convolução

- A correlação mais é a *filtragem de média* que, no caso unidimensional temos que o valor do nível de cinza de um pixel é obtida pela média de seus dois vizinhos
- Seja a imagem unidimensional  $f$  representada pelo vetor:

...	5	4	2	3	7	4	6	5	4	6	...
-----	---	---	---	---	---	---	---	---	---	---	-----

- O cálculo da média para o pixel de valor 3 produzirá o valor 4 que é a média entre 2, 3, e 7. Neste caso considera-se apenas uma janela de tamanho 3
- A filtragem é feita deslocando-se uma máscara de tamanho 3 ( $w = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ ) pela imagem. Em geral, filtros sempre tem tamanho ímpar para produzir um elemento central

### Correlação Unidimensional

$$w.f(x) = \sum_{i=-\lfloor m/2 \rfloor}^{\lfloor m/2 \rfloor} w(i).f(x+i)$$

# Realce de Imagens

## Correlação e Convolução

### Correlação Bidimensional

$$w.f(x, y) = \sum_{i=\lfloor -m/2 \rfloor}^{\lfloor m/2 \rfloor} \sum_{j=\lfloor -n/2 \rfloor}^{\lfloor n/2 \rfloor} w(i, j).f(x + i, y + j)$$

# Realce de Imagens

## Correlação e Convolução

### Correlação Bidimensional

$$w.f(x, y) = \sum_{i=\lfloor -m/2 \rfloor}^{\lfloor m/2 \rfloor} \sum_{j=\lfloor -n/2 \rfloor}^{\lfloor n/2 \rfloor} w(i, j).f(x + i, y + j)$$

### Convolução

$$w * f(x, y) = \sum_{i=\lfloor -m/2 \rfloor}^{\lfloor m/2 \rfloor} \sum_{j=\lfloor -n/2 \rfloor}^{\lfloor n/2 \rfloor} w(i, j) * f(x - i, y - j)$$

# Realce de Imagens

## Correlação e Convolução

### Correlação Bidimensional

$$w.f(x, y) = \sum_{i=\lfloor -m/2 \rfloor}^{\lfloor m/2 \rfloor} \sum_{j=\lfloor -n/2 \rfloor}^{\lfloor n/2 \rfloor} w(i, j).f(x + i, y + j)$$

### Convolução

$$w * f(x, y) = \sum_{i=\lfloor -m/2 \rfloor}^{\lfloor m/2 \rfloor} \sum_{j=\lfloor -n/2 \rfloor}^{\lfloor n/2 \rfloor} w(i, j) * f(x - i, y - j)$$

O filtro de convolução consiste na matriz de correlação rotacionada em 180 graus

# Realce de Imagens

## Correlação e Convolução

- Os pixels na região de borda devem ser tratados de forma diferente por não terem todos os vizinhos

# Realce de Imagens

## Correlação e Convolução

- Os pixels na região de borda devem ser tratados de forma diferente por não terem todos os vizinhos
- Abordagem tradicional: usar filtros com matrizes quadradas  $n \times n$  com  $n$  ímpar com valores pequenos (para eficiência computacional)

# Realce de Imagens

## Correlação e Convolução

- Os pixels na região de borda devem ser tratados de forma diferente por não terem todos os vizinhos
- Abordagem tradicional: usar filtros com matrizes quadradas  $n \times n$  com  $n$  ímpar com valores pequenos (para eficiência computacional)
- Exemplo: para uma máscara  $3 \times 3$  em uma imagem  $512 \times 512$  temos 9 multiplicações e 8 adições para cada pixel, portanto 2.359.596 multiplicações e 2.097.152 adições (**sem considerar ainda as computações para os pixels de borda!!**)

# Realce de Imagens

## Correlação e Convolução: Exemplo

Seja uma região de uma imagem cujos níveis de cinza são de acordo com a matriz abaixo, e o filtro de convolução  $w$ :

137	115	153
177	213	103
115	182	158

-1	0	1
-2	1	2
-1	0	1

# Realce de Imagens

## Correlação e Convolução: Exemplo

Seja uma região de uma imagem cujos níveis de cinza são de acordo com a matriz abaixo, e o filtro de convolução  $w$ :

137	115	153
177	213	103
115	182	158

-1	0	1
-2	1	2
-1	0	1

A correlação para o pixel em questão é dada por:

$$137 \cdot (-1) + 115 \cdot 0 + 153 \cdot 1 + 177 \cdot (-2) + 213 \cdot 1 + 103 \cdot 2 + 115 \cdot (-1) + 182 \cdot 0 + 158 \cdot 1 = 124$$

A convolução é dada por:

$$137 \cdot 1 + 115 \cdot 0 + 153 \cdot (-1) + 177 \cdot 2 + 213 \cdot 1 + 103 \cdot (-2) + 115 \cdot 1 + 182 \cdot 0 + 158 \cdot (-1) = 302$$

# Realce de Imagens

## Correlação e Convolução: Exercício

- Faça um programa que filtre uma imagem P/B a partir da média da vizinhança-8

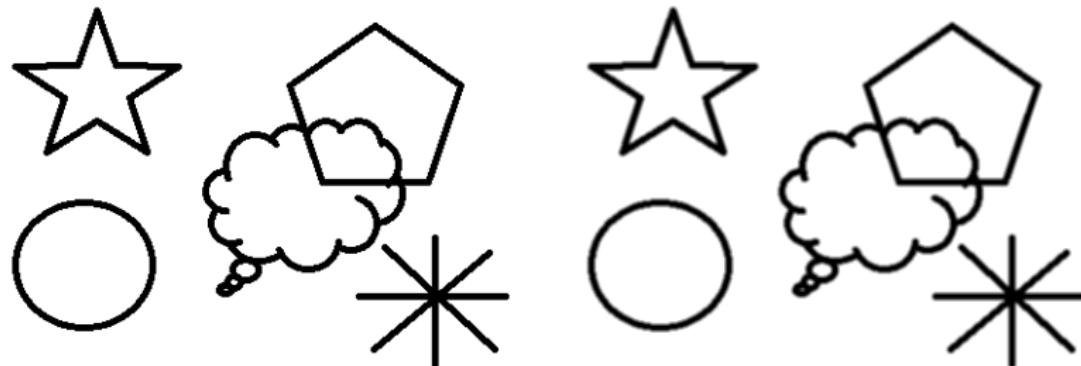
# Realce de Imagens

## Correlação e Convolução: Exercício

```
1 def Smooth(matriz):
2     size = matriz.shape
3
4     H = size[0]
5     W = size[1]
6
7     mascara = np.ones((3,3))
8
9     suavizado = np.zeros((H,W))
10    for x in range(H):
11        for y in range(W):
12            R = 0
13            components = 0
14            for ix in range(-1,2):
15                for iy in range(-1,2):
16                    if x+ix >= 0 and x+ix < H and y+iy >= 0 and y+iy < W:
17                        R = R + mascara[ix][iy] * matriz[x+ix][y+iy]
18                        components += 1
19            suavizado[x][y] = int(round(R/components))
20
return suavizado
```

# Realce de Imagens

Correlação e Convolução: Exercício



# Realce de Imagens

## Filtros Passa-baixa

- Filtros passa-baixa produz o efeito de suavização da imagens dado que as altas frequências que correspondem às transições abruptas são atenuadas

# Realce de Imagens

## Filtros Passa-baixa

- Filtros passa-baixa produz o efeito de suavização da imagens dado que as altas frequências que correspondem às transições abruptas são atenuadas
- Outro efeito desejável dos filtros passa-baixa é a redução do ruído

# Realce de Imagens

## Filtros Passa-baixa

- Filtros passa-baixa produz o efeito de suavização da imagens dado que as altas frequências que correspondem às transições abruptas são atenuadas
- Outro efeito desejável dos filtros passa-baixa é a redução do ruído
- Por outro lado, este tipo de filtro causa o '*borramento*' das imagens e, desta forma, detalhes finos podem ser removidos da imagem

# Realce de Imagens

Filtros Passa-baixa: exemplos

$$h_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$h_2 = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$h_3 = \frac{1}{49} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$h_4 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$h_5 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

# Realce de Imagens

Filtros Passa-baixa: exemplos

- $h_1, h_2,$  e  $h_3$  são denominados **filtros de média** ou **filtros-caixa**

# Realce de Imagens

Filtros Passa-baixa: exemplos

- $h_1, h_2,$  e  $h_3$  são denominados **filtros de média** ou **filtros-caixa**
- Esse filtro suaviza regiões com variações no nível de cinza e preservam regiões constantes

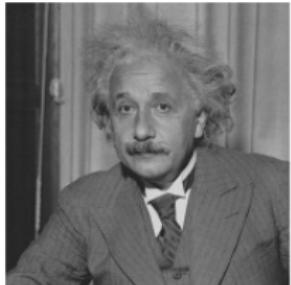
# Realce de Imagens

## Filtros Passa-baixa: exemplos

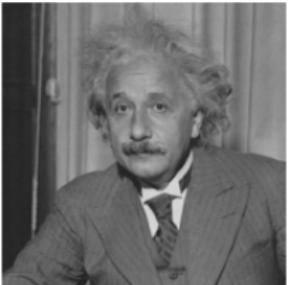
- $h_1, h_2,$  e  $h_3$  são denominados **filtros de média** ou **filtros-caixa**
- Esse filtro suaviza regiões com variações no nível de cinza e preservam regiões constantes
- Os filtros  $h_4, h_5$  representam uma ponderação em função da distância e orientação dos vizinhos

# Realce de Imagens

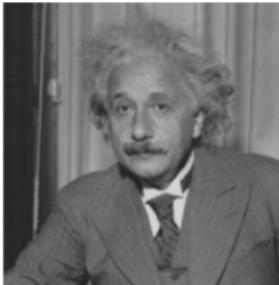
Filtros Passa-baixa: exemplos



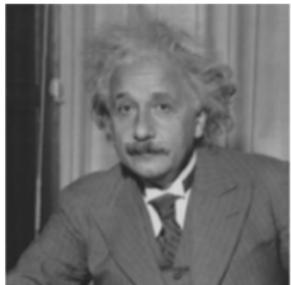
(a) imagem original



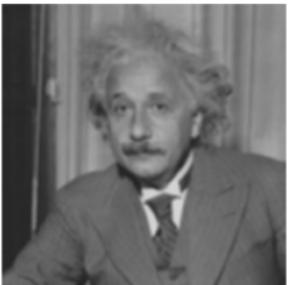
(b) máscara  $3 \times 3$



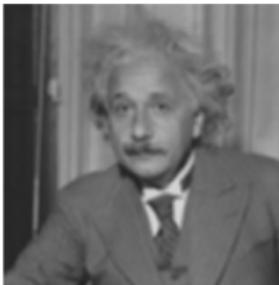
(c) máscara  $5 \times 5$



(d) máscara  $7 \times 7$



(e) máscara  $9 \times 9$



(f) máscara  $11 \times 11$

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

- O processamento com filtros não-lineares procura evitar a suavização homogênea

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

- O processamento com filtros não-lineares procura evitar a suavização homogênea
- Um tipo popular de filtro não-linear são os **filtros estatísticos de ordem**

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

- O processamento com filtros não-lineares procura evitar a suavização homogênea
- Um tipo popular de filtro não-linear são os **filtros estatísticos de ordem**
- Para uma vizinhança contendo  $m$  pixels, estes são ordenados em um conjunto  $p_1, p_2, \dots, p_m$  onde  $p_i \leq p_{i+1} \quad i = 1, 2, \dots, m - 1$

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

- O processamento com filtros não-lineares procura evitar a suavização homogênea
- Um tipo popular de filtro não-linear são os **filtros estatísticos de ordem**
- Para uma vizinhança contendo  $m$  pixels, estes são ordenados em um conjunto  $p_1, p_2, \dots, p_m$  onde  $p_i \leq p_{i+1} \quad i = 1, 2, \dots, m - 1$
- Em imagens monocromáticas, ordena-se em função das intensidades dos pixels. Já nas multibandas, usam-se as cores

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

- Um filtro não-linear bastante importante é o **filtro da mediana** o qual consiste em se substituir a intensidade de cada pixel pela mediana da vizinhança

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

- Um filtro não-linear bastante importante é o **filtro da mediana** o qual consiste em se substituir a intensidade de cada pixel pela mediana da vizinhança
- Para uma vizinhança contendo  $n \times n$  pixels com  $n$  ímpar, a mediana encontra-se na posição  $(n^2 + 1)/2$

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

- Um filtro não-linear bastante importante é o **filtro da mediana** o qual consiste em se substituir a intensidade de cada pixel pela mediana da vizinhança
- Para uma vizinhança contendo  $n \times n$  pixels com  $n$  ímpar, a mediana encontra-se na posição  $(n^2 + 1)/2$
- Este filtro ajuda a reduzir o efeito de ruído impulsivo (sal-e-pimenta)

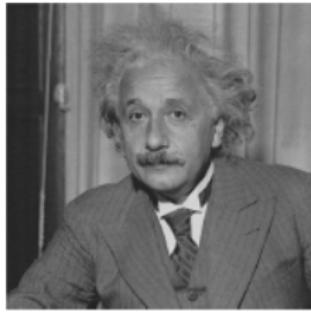
# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

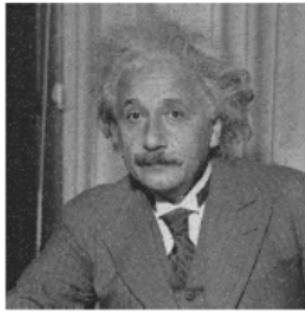
- Um filtro não-linear bastante importante é o **filtro da mediana** o qual consiste em se substituir a intensidade de cada pixel pela mediana da vizinhança
- Para uma vizinhança contendo  $n \times n$  pixels com  $n$  ímpar, a mediana encontra-se na posição  $(n^2 + 1)/2$
- Este filtro ajuda a reduzir o efeito de ruído impulsivo (sal-e-pimenta)
- E ainda, este filtro não introduz valores de níveis de cinza diferentes daqueles contidos na imagem original e, por afetar menos as bordas, pode ser aplicado iterativamente

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

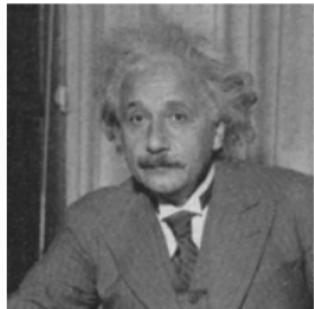


(a) imagem original

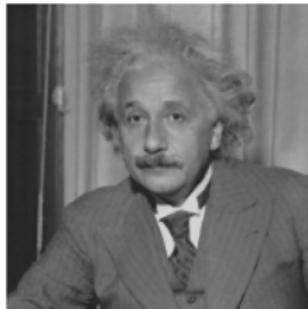


(b) com ruído impul-

sivo



(c) após filtro da  
média  $5 \times 5$



(d) após filtro da me-  
diana  $5 \times 5$

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

- Outros exemplos de filtros não-lineares são: filtro mínimo, máximo e de moda

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

- Outros exemplos de filtros não-lineares são: filtro mínimo, máximo e de moda

**Mínimo** O pixel no centro da máscara é substituído pelo valor mínimo da vizinhança. Desta forma, regiões escuras são aumentadas, dominando as regiões mais claras

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

- Outros exemplos de filtros não-lineares são: filtro mínimo, máximo e de moda

**Mínimo** O pixel no centro da máscara é substituído pelo valor mínimo da vizinhança. Desta forma, regiões escuras são aumentadas, dominando as regiões mais claras

**Máximo** Analogamente, neste filtro o pixel central é substituído pelo valor máximo, dando ênfase nas regiões claras

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

- Outros exemplos de filtros não-lineares são: filtro mínimo, máximo e de moda

**Mínimo** O pixel no centro da máscara é substituído pelo valor mínimo da vizinhança. Desta forma, regiões escuras são aumentadas, dominando as regiões mais claras

**Máximo** Analogamente, neste filtro o pixel central é substituído pelo valor máximo, dando ênfase nas regiões claras

**Moda** seleciona o valor de maior frequência na vizinhança e o substitui no pixel central

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares

15	10	25
20	35	10
35	40	35

original

15	10	25
20	10	10
35	40	35

mínimo

15	10	25
20	40	10
35	40	35

máximo

15	10	25
20	25	10
35	40	35

mediana

15	10	25
20	35	10
35	40	35

moda

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares: Gaussianos

### Filtro Gaussiano

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-(x^2 + y^2)}{2\sigma^2}\right)$$

- 1 É um filtro *isotrópico*, ou seja, é simétrico em relação à rotação, o que produz o mesmo resultado em todas as direções

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares: Gaussianos

### Filtro Gaussiano

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-(x^2 + y^2)}{2\sigma^2}\right)$$

- ① É um filtro *isotrópico*, ou seja, é simétrico em relação à rotação, o que produz o mesmo resultado em todas as direções
- ② A suavização é feita pela substituição de cada pixel pela média ponderada da vizinhança, sendo que o peso de um vizinho decresce monotonicamente em função da distância ao pixel central

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares: Gaussianos

### Filtro Gaussiano

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-(x^2 + y^2)}{2\sigma^2}\right)$$

- 1 É um filtro *isotrópico*, ou seja, é simétrico em relação à rotação, o que produz o mesmo resultado em todas as direções
- 2 A suavização é feita pela substituição de cada pixel pela média ponderada da vizinhança, sendo que o peso de um vizinho decresce monotonicamente em função da distância ao pixel central
- 3 O grau de suavização desse filtro está associado ao parâmetro  $\sigma$ . Quanto maior o seu valor, maior a largura do filtro e a suavização

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares: Gaussianos

### Filtro Gaussiano

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-(x^2 + y^2)}{2\sigma^2}\right)$$

- ➊ É um filtro *isotrópico*, ou seja, é simétrico em relação à rotação, o que produz o mesmo resultado em todas as direções
- ➋ A suavização é feita pela substituição de cada pixel pela média ponderada da vizinhança, sendo que o peso de um vizinho decresce monotonicamente em função da distância ao pixel central
- ➌ O grau de suavização desse filtro está associado ao parâmetro  $\sigma$ . Quanto maior o seu valor, maior a largura do filtro e a suavização
- ➍ Gaussianas são separáveis, ou seja, uma convolução Gaussiana pode ser realizada com um filtro unidirecional aplicado 2x ortogonalmente, o que reduz consideravelmente o número de operações utilizadas

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares: Gaussianos

- Um uso comum de Gaussianas é através da expansão binomial:

$$(a + b)^n = \sum_{k=0}^n \frac{n!}{k!(n-k)!} a^{n-k} b^k$$

- Os coeficientes da expansão binomial podem ser obtidas pelo *Triângulo de Pascal*:

			1								
				1	1	1					
					1	2	1				
						3	3	1			
							1	1			
								1			
									1		
										1	
1	1	4									
1	5	10	10	10	10	10	10	10	10	10	10
1	6	15	20	15	20	15	20	15	20	15	20
1											

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares: Gaussianos

- Uma máscara unidimensional de tamanho  $n$  pode ser obtida pela  $n$ -ésima linha do triângulo de Pascal

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares: Gaussianos

- Uma máscara unidimensional de tamanho  $n$  pode ser obtida pela  $n$ -ésima linha do triângulo de Pascal
- Exemplo:

$$\frac{1}{16} \begin{array}{|c|c|c|c|c|} \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array}$$

é obtida pela quinta linha do triângulo de Pascal dividido pela soma dos coeficientes, ou seja,  $2^{n-1}$

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares: Gaussianos

- Uma máscara unidimensional de tamanho  $n$  pode ser obtida pela  $n$ -ésima linha do triângulo de Pascal
- Exemplo:

$$\frac{1}{16} \begin{array}{|c|c|c|c|c|} \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array}$$

é obtida pela quinta linha do triângulo de Pascal dividido pela soma dos coeficientes, ou seja,  $2^{n-1}$

- O desvio padrão do filtro Gaussiano é dado por:

$$\sigma = \frac{\sqrt{n - 1}}{2}$$

# Realce de Imagens

## Filtros Passa-baixa Não-Lineares: Gaussianos

- Para o caso bidimensional, uma máscara para o filtro Gaussiano com  $\sigma = 1$ , é obtida por duas máscaras unidimensionais (horizontal e vertical):

$$\frac{1}{16} \begin{array}{|c|c|c|c|c|} \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array} = \frac{1}{256} \quad (4)$$

$\frac{1}{16}$        $\frac{1}{16}$        $\frac{1}{16}$

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	19	4
1	4	6	4	1

# Realce de Imagens

Filtros Passa-baixa Não-Lineares: Gaussianos

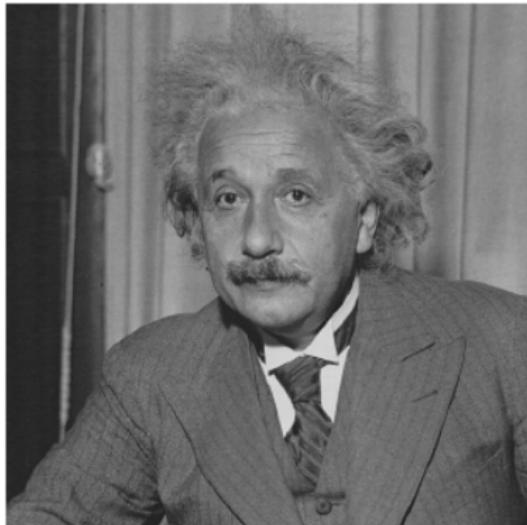


Imagen Original

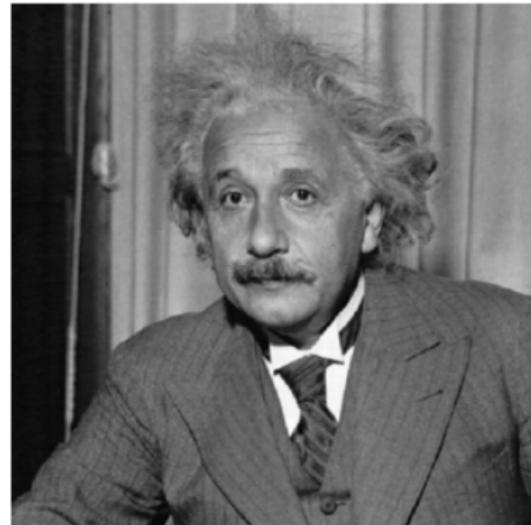


Imagen suavizada por filtro Gaussiano

# Realce de Imagens

## Filtragem com Preservação de Bordas

- Filtros passa-baixa para redução de ruído podem suprimir linhas finas ou cantos de objetos

# Realce de Imagens

## Filtragem com Preservação de Bordas

- Filtros passa-baixa para redução de ruído podem suprimir linhas finas ou cantos de objetos
- Motivo: tais filtros não levam em consideração se um pixel está localizado sobre uma borda ou se seus vizinhos apresentam uma orientação específica

# Realce de Imagens

## Filtragem com Preservação de Bordas

- Filtros passa-baixa para redução de ruído podem suprimir linhas finas ou cantos de objetos
- Motivo: tais filtros não levam em consideração se um pixel está localizado sobre uma borda ou se seus vizinhos apresentam uma orientação específica
- As figuras (a) e (b) abaixo ilustram regiões contendo uma linha vertical e um canto de objeto, tal que a aplicação de um filtro da mediana causaria a supressão de detalhes

0	1	0
0	1	0
0	1	0

(a)

0	0	0
1	1	0
1	1	0

(b)

# Realce de Imagens

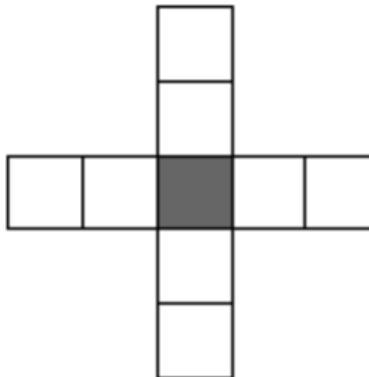
## Filtragem com Preservação de Bordas

- Máscaras não retangulares podem ser utilizadas para reduzir o efeito da supressão de determinados detalhes da imagem

# Realce de Imagens

## Filtragem com Preservação de Bordas

- Máscaras não retangulares podem ser utilizadas para reduzir o efeito da supressão de determinados detalhes da imagem
- O exemplo a seguir, centrado em um pixel de interesse, pode ser utilizado para preservar linhas horizontais e verticais na filtragem mediana

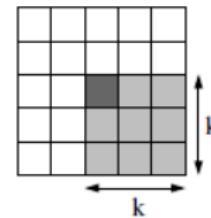
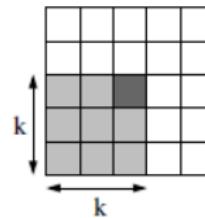
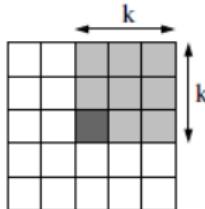
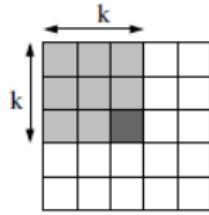


# Realce de Imagens

## Filtragem com Preservação de Bordas

### Máscara de Kuwahara

Considera uma região quadrada  $(2k - 1)^2$  ao redor do pixel  $(x, y)$   
Essa região é subdividida em quatro janelas de  $k \times k$  pixels



# Realce de Imagens

Filtragem com Preservação de Bordas

- A variância dos níveis de cinza é calculada para cada janela

# Realce de Imagens

## Filtragem com Preservação de Bordas

- A variância dos níveis de cinza é calculada para cada janela
- O valor de cada pixel  $(x, y)$  é substituído pela média dos níveis de cinza da janela de variância mínima

# Realce de Imagens

## Filtragem com Preservação de Bordas

- A variância dos níveis de cinza é calculada para cada janela
- O valor de cada pixel  $(x, y)$  é substituído pela média dos níveis de cinza da janela de variância mínima
- A ideia é que regiões que contém bordas possuem variância mais alta do que regiões homogêneas

# Realce de Imagens

## Filtragem com Preservação de Bordas

- A variância dos níveis de cinza é calculada para cada janela
- O valor de cada pixel  $(x, y)$  é substituído pela média dos níveis de cinza da janela de variância mínima
- A ideia é que regiões que contém bordas possuem variância mais alta do que regiões homogêneas

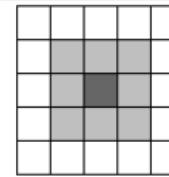
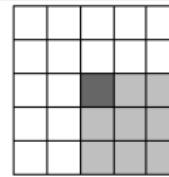
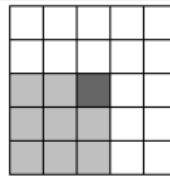
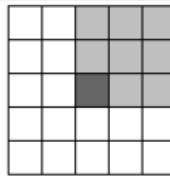
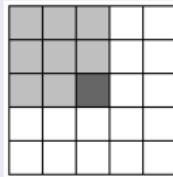
### Algoritmo

- ① **for** cada pixel  $f(x, y)$  **do**
- ②   calcular variância de cada máscara do conjunto sobre o pixel  $f(x, y)$
- ③   escolher a máscara de variância mínima
- ④   atribuir ao pixel  $f(x, y)$  a intensidade média da máscara selecionada

# Realce de Imagens

Filtragem com Preservação de Bordas: variantes

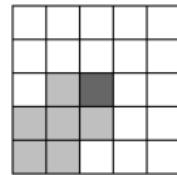
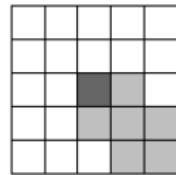
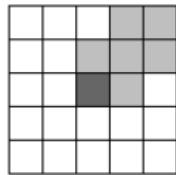
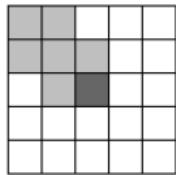
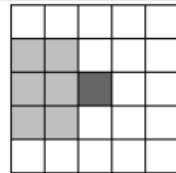
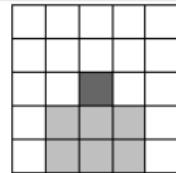
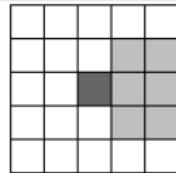
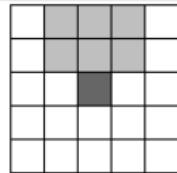
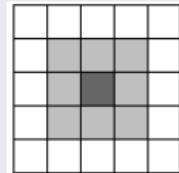
## Máscaras de Tomita e Tsuji



# Realce de Imagens

Filtragem com Preservação de Bordas: variantes

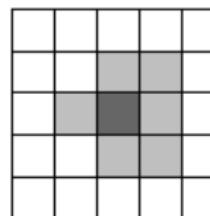
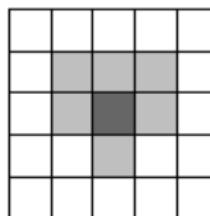
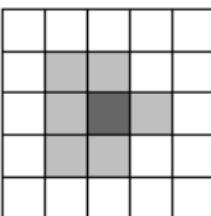
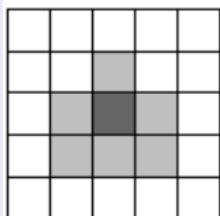
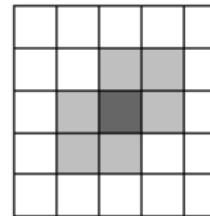
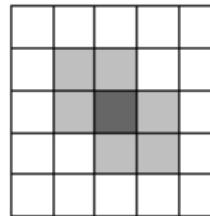
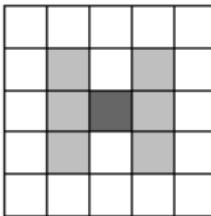
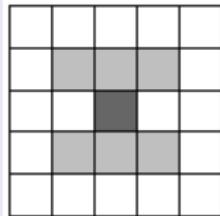
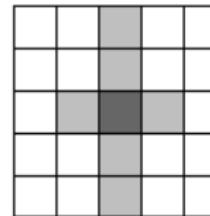
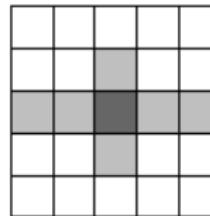
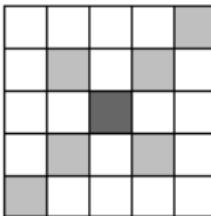
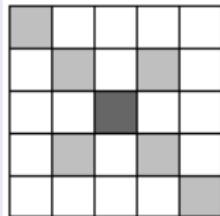
## Máscaras de Nagao e Matsuyama



# Realce de Imagens

Filtragem com Preservação de Bordas: variantes

## Máscaras de Somboonkaew



# Segmentação de Imagens

# Morfologia Matemática

# Próxima Seção

Em Construção [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]