



Universidade de Brasília  
Instituto de Ciências Exatas  
Departamento de Matemática  
Programa de Mestrado Profissional  
em Matemática em Rede Nacional



# **Introdução ao R no Ensino Médio Usando Recuperação de Imagens como Motivação**

Kellem Corrêa Santos

Brasília

2020

Kellem Corrêa Santos

# **Introdução ao R no Ensino Médio Usando Recuperação de Imagens como Motivação**

Dissertação apresentada ao Departamento de Matemática da Universidade de Brasília, como parte dos requisitos do “Programa” de Mestrado Profissional em Matemática em Rede Nacional - PROFMAT, para obtenção do grau de Mestre.

Universidade de Brasília - UnB  
Departamento de Matemática - MAT  
PROFMAT - SBM

Orientador: Prof. Dr. Vinicius de Carvalho Rispoli

Brasília  
2020

Ficha catalográfica elaborada automaticamente,  
com os dados fornecidos pelo(a) autor(a)

CS237i      Corrêa Santos, Kellelem  
Introdução ao R no Ensino Médio Usando Recuperação de  
Imagens como Motivação / Kellelem Corrêa Santos; orientador  
Vinicius de Carvalho Rispoli. -- Brasília, 2020.  
111 p.

Dissertação (Mestrado - Mestrado Profissional em  
Matemática) -- Universidade de Brasília, 2020.

1. Inpainting. 2. Programação. 3. R. 4. Ensino Médio. 5.  
PROFMAT. I. de Carvalho Rispoli, Vinicius, orient. II.  
Título.

Universidade de Brasília  
Instituto de Ciências Exatas  
Departamento de Matemática

# Introdução ao R no Ensino Médio Usando Recuperação de Imagens como Motivação

por

Kellem Corrêa Santos

*Dissertação apresentada ao Departamento de Matemática da Universidade de Brasília,  
como parte dos requisitos “Programa” de Mestrado Profissional em Matemática em  
Rede Nacional – PROFMAT, para obtenção do grau de*

MESTRE EM MATEMÁTICA

Brasília, 29 de julho de 2020.

Comissão Examinadora:

---

Prof. Vinicius de Carvalho Rispoli (Orientador)

---

Prof. Marcus Vinicius Chaffim Costa– FGA/UnB

---

Prof. Matheus Bernardini de Souza- FGA/UnB

*À minha mãe Maria Vilma, que é meu tesouro, quem me dá forças e ânimo para continuar a viver e sem a qual eu não realizaria tantos feitos. Mamãe, você é meu exemplo e minha inspiração de sempre; sou o que sou graças à sua dedicação, muito obrigada! Te amo!*

# Agradecimentos

Em primeiro lugar, gostaria de agradecer a Deus pela minha vida, pelas oportunidades que Ele me proporcionou, pela família linda que Ele me deu e por estar sempre comigo me dando forças para seguir em frente mesmo diante dos obstáculos.

À minha mãe Maria Vilma, meu tesouro, minha vida, minha maior encorajadora. Agradeço de coração a dedicação incansável, o apoio, a paciência e a compreensão em todos os momentos da minha vida. Obrigada por você existir na minha vida e por ser essa mãe gigante que você é.

A absolutamente todos os meus professores, pelos ensinamentos (não só técnicos, mas também passados com exemplos de vida e conversas nos momentos em que precisei). Vocês são os grandes responsáveis pela minha formação. Muito obrigada.

Ao meu orientador Prof. Vinicius. Obrigada pela enorme paciência, pela dedicação, pela orientação precisa e oportuna em todas as etapas deste trabalho.

Aos colegas do PROFMAT, que turma maravilhosa! Obrigada pela parceria – inclusive nas madrugadas com as listas e os estudos de prova – bem como pela amizade fora de sala. Em especial aos amigos Bernado, Dema e Lucas: obrigada pelo apoio, pelas risadas e por sempre alegrarem meu dia!

À UnB, ao Departamento de Matemática e à coordenação do PROFMAT-UnB, muito obrigada pela oportunidade de participar desse curso e aprender tanto com um corpo técnico tão qualificado, é uma honra ter chance de conhecer e estudar com esses mestres!

Ao Banco Central do Brasil, em especial ao Departamento Pessoal (Depes) e ao Departamento de Reservas Internacionais (Depin), onde trabalho, agradeço a compreensão e o apoio que foram fundamentais para a realização desse curso de mestrado.

*“Mathematics is the language in which the gods talk to people”*  
*Platão*

# Resumo

Este trabalho tem por objetivo apresentar roteiros de aula para alunos do Ensino Médio utilizando recursos computacionais e a técnica de *inpainting* por meio da linguagem de programação R. Para tal, serão apresentados a linguagem de programação R e o conceito de *inpainting*. Após isso, será apresentada a biblioteca (com seu código-fonte em R) que implementa a técnica de *inpainting* de forma básica usando o filtro Gaussiano. Então, será descrito um código exemplo de como utilizá-la. Os roteiros de aula incluem apresentação da linguagem R de programação e a teoria básica de filtro Gaussiano aos alunos bem como exemplos de códigos-fonte, além de introdução à técnica de *inpainting* e programação por parte dos alunos – utilizando a biblioteca – para implementarem a técnica em aula.

**Palavras-chaves:** *Inpainting*. Programação. R. Ensino Médio.



# Abstract

This work aims to present class scripts for high school students including computational resources and the inpainting technique using the R programming language. For this purpose, the R programming language and the inpainting concept will be presented. After that, the library - with its source code in R - will be presented, which implements the inpainting technique in a basic approach using Gaussian filter. Then, an example code of how to use it will be described. Class scripts include presentation of the R programming language and basics of Gaussian filter to students as well as source code examples, introduction to the inpainting technique, presentation of mobile applications that use the technique and programming by students – using the library – to implement the technique in class.

**Key-words:** Inpainting. Programming. R. High School.

# Lista de ilustrações

Figura 2.1 – Tela Inicial do RStudio . . . . .	35
Figura 2.2 – ExemploR.r em R . . . . .	37
Figura 2.3 – Execução do ExemploR.r pelo menu <b>Run</b> . . . . .	38
Figura 2.4 – Execução do ExemploR.r pelo atalho Ctrl + Enter . . . . .	38
Figura 3.1 – Restauração manual de pintura 1 . . . . .	47
Figura 3.2 – Restauração manual de pintura 2 . . . . .	47
Figura 3.3 – Restauração manual de pintura 3 . . . . .	48
Figura 3.4 – Recuperação de imagens feitas por Bertalmio . . . . .	50
Figura 3.5 – Imagem com ruído “Sal-e-Pimenta” antes e depois de sua remoção por filtro da mediana . . . . .	54
Figura 3.6 – Processo de filtragem espacial linear usando máscara $3 \times 3$ . . . . .	67
Figura 3.7 – Função Gaussiana de duas variáveis com $\sigma = 1$ . . . . .	70
Figura 3.8 – Comando no Console para instalação da biblioteca <b>imager</b> . . . . .	73
Figura 3.9 – Janela de progresso de <i>download</i> de pacotes que a biblioteca <b>imager</b> utiliza . . . . .	73
Figura 3.10–Console após a completa instalação da biblioteca <b>imager</b> . . . . .	73
Figura 3.11–Imagem <b>boats</b> . . . . .	74
Figura 3.12–Resultado da aplicação do filtro Gaussiano na imagem <b>boats</b> com falhas espalhadas . . . . .	76
Figura 3.13–Resultado da aplicação do filtro Gaussiano na imagem <b>boats</b> com falhas em traços . . . . .	77
Figura 3.14–Resultado da aplicação do filtro Gaussiano na imagem <b>boats</b> com falha em bloco . . . . .	77
Figura 3.15–Resultado da aplicação do filtro Gaussiano na imagem <b>boats</b> com falha em bloco mais largo . . . . .	78
Figura A.1–Tabela com métodos baseados em sequências . . . . .	107
Figura A.2–Tabela com métodos baseados em redes neurais convolucionais . . . . .	108
Figura A.3–Esquema de funcionamento das redes GANs . . . . .	109

# Sumário

	<b>Introdução</b>	<b>12</b>
<b>1</b>	<b>TECNOLOGIAS DE INFORMAÇÃO E COMUNICAÇÃO NA EDUCAÇÃO MATEMÁTICA</b>	<b>14</b>
1.1	Tecnologias na Educação	14
1.2	Ambientes Virtuais de Aprendizagem	15
1.3	Educação a Distância	17
1.3.1	Histórico	17
1.3.2	EAD no Brasil	20
1.4	Redes Sociais	23
1.5	Aplicativos	25
1.5.1	GeoGebra	25
1.5.2	Rei da Matemática	27
1.6	Prós e Contras	29
<b>2</b>	<b>A LINGUAGEM DE PROGRAMAÇÃO R</b>	<b>32</b>
2.1	A Linguagem	32
2.2	Utilizando o R	34
2.2.1	Operações Básicas	38
2.2.2	Atribuição de Valores a Variáveis	39
2.2.3	Vetores	40
2.2.4	Matrizes	41
<b>3</b>	<b>INPAINTING</b>	<b>46</b>
3.1	O <i>Inpainting</i>	46
3.2	História	49
3.3	Tipos de Técnicas de <i>Inpainting</i>	50
3.3.1	<i>Inpainting</i> Estrutural	50
3.3.2	<i>Inpainting</i> Textural	51
3.3.3	<i>Inpainting</i> Estrutural e Textural	52
3.4	Ruídos	53
3.5	Recuperação de Imagens	55
3.5.1	Equação do Calor e Recuperação de Imagens	55
3.5.2	Precisão do Método	58
3.6	Filtragem	64
3.6.1	Filtragem Digital	66

3.6.2	Filtro Gaussiano . . . . .	69
<b>3.7</b>	<b>Implementação em R . . . . .</b>	<b>72</b>
<b>4</b>	<b>ROTEIROS DE AULA . . . . .</b>	<b>79</b>
<b>4.1</b>	<b>Breve Ensino da Linguagem R . . . . .</b>	<b>79</b>
4.1.1	Aula 1: A linguagem R e o RStudio . . . . .	80
4.1.2	Aula 2: Variáveis e Funções . . . . .	81
4.1.3	Aula 3: Uso de Bibliotecas . . . . .	82
<b>4.2</b>	<b>Exemplos de Aplicações em R . . . . .</b>	<b>84</b>
4.2.1	Aula 4: Cálculo Aritmético em R . . . . .	84
4.2.2	Aula 5: Recorrências em R . . . . .	85
4.2.3	Aula 6: Manipulação de Arquivos e Imagens em R . . . . .	87
<b>4.3</b>	<b>Inserção de Falhas na Imagem . . . . .</b>	<b>88</b>
4.3.1	Aula 7: Elementos da Imagem . . . . .	89
4.3.2	Aula 8: Padrões e Ruídos da Imagem . . . . .	90
4.3.3	Aula 9: Inserção de Falhas na Imagem em R . . . . .	91
<b>4.4</b>	<b>Breve Explicação sobre Filtro Gaussiano . . . . .</b>	<b>92</b>
4.4.1	Aula 10: Filtro de Imagem . . . . .	93
4.4.2	Aula 11: Máscara . . . . .	94
4.4.3	Aula 12: Filtro Gaussiano . . . . .	95
<b>4.5</b>	<b>Recuperação da Imagem . . . . .</b>	<b>97</b>
4.5.1	Aula 13: Recuperação da Imagem em R . . . . .	97
	<b>Considerações Finais . . . . .</b>	<b>99</b>
	<b>Referências . . . . .</b>	<b>101</b>
	<b>APÊNDICE A – TÉCNICAS DE RECUPERAÇÃO DE IMAGENS .</b>	<b>105</b>
<b>A.1</b>	<b>Outras Abordagens de <i>Inpainting</i> . . . . .</b>	<b>105</b>
<b>A.2</b>	<b><i>Compressive Sensing</i> . . . . .</b>	<b>109</b>

# Introdução

O currículo do Ensino Básico não inclui matérias de Computação. Com o mundo globalizado torna-se interessante abordar o tema com os alunos do Ensino Médio, apresentando a eles algumas das inúmeras aplicações da Programação existentes.

Faz parte de muitos desses alunos o uso do celular e dos aplicativos *mobile*. Também fazem parte de seu cotidiano as fotografias. Por ser um tema de conhecimento dos jovens do Ensino Médio e que venha a despertar o seu interesse, o *inpainting* foi escolhido neste trabalho por apresentar reconstrução de imagens com pedaços faltantes.

Há aplicativos de celular que fazem esse procedimento e os alunos podem mostrar grande interesse ao entender como isso funciona e ao implementar o método. Assim, é apresentada uma linguagem de programação (R) em sala de aula juntamente a uma aplicação que faz parte do mundo e do dia-a-dia desses jovens.

Este trabalho visa apresentar roteiros de aula para alunos do Ensino Médio – de qualquer um dos três anos – com o tema de *inpainting* utilizando programação na linguagem R.

Dessa forma, são apresentados: conceitos de Programação, a linguagem R (na qual será implementada a técnica de *inpainting* com os alunos em sala), a técnica de reconstrução de imagens e os roteiros de aula para aplicação em sala. No final, aos professores que desejarem continuar o trabalho e avançar no ensino de Computação e de reconstrução de imagens, são feitas sugestões de trabalho futuro.

Para tanto, no primeiro capítulo é abordado o uso de tecnologias no âmbito do Ensino Básico, mais especificamente o uso de tecnologias de informação e comunicação em aulas de Matemática, com possibilidades, limitações, desafios, como os estudiosos vêem o assunto, principais usos e suas implicações. Durante o capítulo, são exemplificadas experiências no Ensino Médio com o auxílio desses recursos.

Em seguida, o segundo capítulo aborda com detalhes a linguagem de programação em R, com exemplos de aplicações e respectivos códigos-fonte. É nesse capítulo que se discute em que é utilizado o R, como é estruturado, suas vantagens, desvantagens, principais funções e como é feito um programa em R, bem como instruções de instalação do programa RStudio (seu compilador) e de suas bibliotecas.

O terceiro capítulo traz descrição, história, vantagens e aplicações da técnica de reconstrução de imagens chamada *inpainting*.

Ainda no terceiro capítulo é apresentada a biblioteca **imager** em R que realiza a reconstrução de imagens utilizando a técnica de *inpainting*, bem como o programa

específico a ser desenvolvido em sala com alunos do Ensino Médio. Esse programa recebe uma foto de entrada, insere algumas faltas de *bits* nela e, posteriormente, a imagem é reconstruída.

O quarto capítulo contém os roteiros de aula para o ensino da linguagem de programação R e da técnica de *inpainting*. Também são incluídos nos roteiros presentes nesse capítulo recursos requeridos, metodologia de ensino e, quando for o caso, sugestão de avaliação. Nele há cada roteiro com o tempo sugerido de aula e o passo a passo para aplicação em sala de aula.

Por fim, no último capítulo é feita uma revisão dos tópicos abordados, bem como uma sugestão de trabalho futuro para os professores que desejarem avançar no ensino da Programação e da técnica de reconstrução de imagens.

## Metodologia

Este trabalho busca explicar o conteúdo aos professores de Ensino Médio que tenham interesse em lecionar algumas aulas extra com seus alunos sobre o tema computacional de maneira básica e que desperte o interesse deles.

Para atingir esse fim, primeiramente foram apresentadas as tecnologias de informação e comunicação – as chamadas TICs – em conjunto com seus prós e contras no processo de aprendizagem.

Esses conceitos pavimentaram o terreno para os capítulos seguintes, que abordam a linguagem de programação R – a qual será utilizada para fins didáticos com os alunos – e uma técnica de reconstrução de imagens, o chamado *Inpainting*.

Por fim, os roteiros de aula são apresentados como sugestão de aplicação em sala de aula no âmbito do Ensino Médio. Com isso, busca-se oportunizar o ensino da introdução à técnica de *inpainting* por meio de programação em R com algumas aulas.

# 1 Tecnologias de Informação e Comunicação na Educação Matemática

Neste capítulo será abordado o uso da tecnologia, em especial das Tecnologias de Informação e Comunicação (TICs), e dos recursos computacionais na Educação Matemática, com possibilidades, limitações, desafios, como os estudiosos vêem o assunto, principais usos e suas implicações. Durante o capítulo, são exemplificadas experiências no Ensino Médio com o auxílio desses recursos.

## 1.1 Tecnologias na Educação

O uso da tecnologia na educação vem sendo investigado há algumas décadas. O MIT (*Massachusetts Institute of Technology*) iniciou o estudo nos anos 60 e chegou a desenvolver um ambiente de programação, especialmente utilizado no ensino da Geometria, a exemplo do GeoGebra – muito utilizado nos dias de hoje.

Nos últimos anos, principalmente com o avanço da Computação, as pesquisas na área têm aumentado exponencialmente. Junto a elas, surgem *softwares* de apoio, formações e especializações de professores, bem como treinamento continuado, de forma a tornar possível a aplicação das investigações e das ideias que surgem para incluir a tecnologia no ensino.

Além disso, cada vez mais os educadores preocupam-se em implantar o seu uso em suas práticas, principalmente nos tempos atuais nos quais o mundo globalizado exige tempestividade e os alunos conectam-se a diversas fontes de informação em tempo real.

As tecnologias representam uma oportunidade para mudanças na educação, principalmente da prática docente. Esta começa a mudar o foco, que antes concentrava-se no professor e agora começa a ser centrada nos alunos, com o propósito de atender os anseios e as demandas de conhecimento deles que possuem acesso a uma gama de informação. Por causa disso, o papel do educador é essencial, em especial no sentido de direcionar os alunos ao conhecimento – e não apenas à informação, disponível em massa na internet.

Uma das vantagens do advento das tecnologias é que ele trouxe aos pesquisadores e educadores a reflexão sobre a educação frente às modificações pelas quais a sociedade tem passado em decorrência da crescente inserção das tecnologias no dia-a-dia do ser humano. As tecnologias ampliam as possibilidades e fornecem diversas formas não convencionais de ensino (por exemplo, hoje é possível, no Brasil, assistir a aulas das melhores universidades internacionais através da internet). Com isso, ideias que antes só ficavam no papel por

limitações de custo, tempo ou recursos físicos tornam-se factíveis com o uso da tecnologia.

Ao mesmo tempo que a tecnologia facilita o ensinar e o aprender, ela também representa um desafio, uma vez que desestabiliza os métodos tradicionais e obriga os profissionais de educação a se adaptarem rapidamente às novidades e às novas formas que vão surgindo a todo tempo, em busca de uma maneira de ensino potencialmente mais eficiente que a praticada.

A velocidade do avanço tecnológico, muitas vezes, faz com que se investiguem as novas modalidades de técnicas de educação antes mesmo que se possam aprofundar estudos acerca do uso de determinada tecnologia. A pressão da sociedade sobre escolas e professores em aplicar técnicas mais tecnológicas e inovadoras acabam por, às vezes, desestabilizarem o profissional e deixá-lo confuso, já que sente-se obrigado a usar determinado *software*, por exemplo, sem embasamento de sua eficácia. O que ocorre é a constatação empírica após a aplicação da técnica sem prévias discussão e reflexão na comunidade da educação.

Os dados empíricos de determinada tecnologia aplicada em sala de aula dependem da percepção do professor, do ambiente escolar, dos recursos computacionais e físicos disponíveis e do contexto dos alunos. Com isso, nem sempre uma tecnologia pode ser bem sucedida em todos os locais de ensino, mesmo que em uma mesma série. Desta forma, quase sempre comprova-se a eficácia – ou não – da aplicação em sala de determinada tecnologia apenas após a tentativa e o seu sucesso – ou não.

Existem diversas opiniões a respeito da implantação de tecnologias no ensino, de que técnicas são mais eficazes e do que deve ou não ser ensinado pelo professor em sala de aula (e o que deve ser ensinado via sistema); essa é uma discussão relevante que continua sem consenso. O importante é que os conteúdos curriculares sejam respeitados e ensinados aos alunos de forma aprofundada, seja por uma estratégia pedagógica ou outra. Vale ressaltar que não se considera a tecnologia capaz de substituir o docente, mas sim auxiliá-lo na transmissão do conhecimento.

A educação, que agora conta com a tecnologia, deve reforçar o papel do aluno como sujeito principal de sua aprendizagem, permitindo-lhe novas descobertas e fornecendo-lhe meios para compartilhá-las com os colegas e professores. Cabe à instituição de ensino desenvolver a aprendizagem de acordo com os quatro pilares da educação definidos pela Unesco (Aprender a Conhecer, Aprender a Fazer, Aprender a Conviver, Aprender a Ser).

## 1.2 Ambientes Virtuais de Aprendizagem

O ambiente escolar atual se desenha como um local propício para o desenvolvimento de atividades por meio da tecnologia. Esse terreno se demonstra fértil à medida



que, para o professor, a tecnologia se revela uma aliada na execução das atividades administrativas inerentes à profissão, como o fechar as notas do bimestre, conferir as faltas, preencher histórico escolar, e até mesmo lecionar; e para o aluno, os meios tecnológicos se configuram como ferramenta de estudo, pois facilitam e complementam a fixação do conteúdo dado em sala de aula, além de promover a integração e a aprendizagem colaborativa.

A aprendizagem colaborativa (como os fóruns de internet) vem para se contrapor à aula exclusivamente expositiva, uma vez que esta última não favorece a integração nem satisfaz as expectativas dos estudantes. Nesse sentido, e imersos no contexto do advento da tecnologia e suas facilidades, estão os Ambientes Virtuais de Aprendizagem (AVA).

Os AVA atuam na dinamização das práticas de ensino e aprendizagem, uma vez que colocam o conteúdo ministrado à disposição do aluno por meios eletrônicos. Além disso, educadores e educandos podem desenvolver seus potenciais lúdicos e criativos, devido à redução dos espaços restritos a professores e alunos. Com isso, os AVA se estabelecem como um lugar comum. A cooperação virtual vem apoiar o processo de desenvolvimento cognitivo e social visando à construção coletiva de conhecimentos, pelo compartilhamento de experiências, informações, dúvidas, desejos e descobertas em tempo real (NOVELLO, 2006). Um AVA pode ser usado também como ferramenta para EAD (Educação a Distância), sendo usado, também, para complementar aulas presenciais com conteúdos virtuais.

Conceitualmente, AVA, para MCKIMM; JOLLIE; CANTILLON, 2003, consiste em um conjunto de ferramentas eletrônicas voltadas ao processo de ensino e aprendizagem. Os principais componentes incluem sistemas que podem organizar conteúdos, acompanhar atividades, além de fornecer ao aluno suporte *online* e comunicação eletrônica.

Para MILLIGAN, 1999, o termo AVA deve ser usado para descrever um aplicativo baseado em um servidor e modelado para gerenciar e administrar os variados aspectos da aprendizagem, como disponibilizar conteúdos ou avisos, acompanhar o aluno, avaliar o processo de ensino, entre outros. No entanto, o autor comenta que, embora existam diversos pacotes com o objetivo de controlar todo o processo de ensino, não se deve excluir o uso de ferramentas individualizadas, cada uma com seu propósito, de forma a criar um ambiente virtual de aprendizagem mais flexível e que atenda aos anseios do docente.

Assim, a depender do caso, é possível escolher um AVA ou outro, ou mesmo agregar mais de um. Diante disso, a definição de AVA deve ser ampla, considerando não somente os programas de AVA prontos, mas também toda e qualquer criação de ambientes baseados em ferramentas agregadas.

De acordo com MILLIGAN, 1999, para a gestão do aprendizado e a disponibilização de materiais, um AVA deve apresentar algumas ferramentas como:

- Controle de acesso

- Controle de tempo:
- Administração
- Avaliação
- Comunicação
- Espaço privativo
- Gerenciamento de recursos
- Apoio
- Manutenção

Alguns AVA disponíveis são: TelEduc, Blackboard, Solar, Sócrates, Amadeus, Dokeos, Sílabo, LMS Estúdio, AulaNet, Moodle e E-Proinfo. Um dos mais populares é o Moodle, principalmente por ser gratuito, escalável e altamente personalizável.

Para o professor, o uso do AVA é um recurso que traz as funcionalidades de uma solução tecnológica capaz de propiciar agilidade e dinamismo na apresentação de conteúdos que estarão à disponibilidade dos alunos em um meio de fácil uso e acesso. A adoção de novas práticas como o AVA traz benefícios para a prática pedagógica, além de uma maior eficiência no trato das atividades administrativas relativas à sua função.

## 1.3 Educação a Distância

A Educação a Distância, mais conhecida como EAD, é uma modalidade de ensino na qual o professor ensina aos alunos fora de sala de aula, separados geograficamente. A modalidade se caracteriza principalmente pela flexibilidade de lugar e horário em que podem ser ministradas e assistidas as aulas. É resultado direto do desenvolvimento das chamadas TICs aplicadas ao processo de ensino.

Com a EAD, docentes e discentes “encurtam” a distância entre si, podendo, inclusive, interagir em tempo real através da comunicação síncrona, simulando o ambiente de sala de aula. Também pode ser feita com uso de comunicação assíncrona, na qual a interação não é em tempo real, por meio de troca de mensagens e materiais.

### 1.3.1 Histórico

A EAD surgiu para suprir a necessidade das pessoas que precisavam estudar e não conseguiam disponibilidade de horário para frequentar as aulas presenciais.

Ao longo do tempo, as metodologias de ensino a distância se modificaram e se modernizaram. Estudiosos, como GARRISON, 1985 e NIPPER, 1989, identificam três gerações nessa área:

1. Primeira Geração: até a década de 60, também chamada de geração textual ou por correspondência, utilizava somente textos impressos enviados pelos Correios
2. Segunda Geração: entre as décadas de 60 e 80, também chamada de geração analógica ou de telecomunicações, utilizava textos impressos juntamente a recursos tecnológicos audiovisuais
3. Terceira Geração: da década de 80 até hoje, também chamada de geração digital ou de telemática, utiliza recursos de TICs disponíveis, em especial a internet

A primeira geração foi marcada pelo envio de material – como notas de aula e exercícios – por correspondência através dos Correios. Uma dificuldade da época era a falta de um tutor, que é o professor disponível para sanar dúvidas e orientar os alunos. Mesmo com o avanço da tecnologia, é comum o uso de material didático impresso nos dias de hoje, em conjunto com a tutoria para dar suporte aos alunos e contribuir para o sucesso do curso.

A segunda geração foi marcada pela utilização de telefone, rádio e televisão. Os textos continuavam a ser enviados pelos Correios, porém recebiam complementação de recursos audiovisuais para resolver exercícios e esclarecer o conteúdo. Os alunos, normalmente, conseguiam tirar dúvidas por telefone com um tutor.

Ainda no contexto da segunda geração de metodologias na EAD, seguem algumas informações sobre cada recurso audiovisual (não necessariamente que foi inventado durante a segunda geração) tido como inovação no ensino entre as décadas de 60 e 80:

- rádio: inventado em 1892, começou a ser usado na EAD no Canadá com a vantagem de atingir grandes distâncias regionais
- telefone: principal meio da época de interação entre professor, coordenador e aluno, era comum uma linha gratuita ou de baixo custo para esse objetivo
- televisão: com o auxílio de vídeo e áudio, foi uma mídia de extrema importância para a EAD, contando com a desvantagem de não permitir interação entre professor e aluno, o que é resolvido com o uso combinado com outras ferramentas
- fita-cassete: trouxe a possibilidade de se ouvir as gravações das aulas repetidas vezes, pausar e retornar para que o entendimento ficasse ainda mais claro

- vídeo VHS: da mesma forma que a fita-cassete, trouxe a possibilidade de rever o vídeo sem limites e em qualquer horário

O mundo vive hoje a terceira geração de EAD, marcada pelo uso de *softwares*. O grande diferencial dessa geração é a facilidade de interação entre o aluno, o professor e o ambiente (virtual) de ensino. Neste modelo, o professor ou tutor passa a atuar como mediador, responsável não só por ensinar, mas também por parametrizar o ambiente virtual de aprendizagem. A mediação pode ser feita de diversas formas. As mais comuns são listadas a seguir.

- *chats*: salas de bate-papo baseadas em trocas de mensagens, normalmente com dois tutores para sanar as dúvidas que os alunos tiverem e também iniciarem o debate
- *audio chats* ou audioconferência: parecida com a anterior, substituindo as mensagens escritas por faladas
- fóruns: plataforma virtual em que os envolvidos discutem um tema por meio de mensagens visualizadas por todos
- *wikis*: páginas da internet que servem para expor determinado tema de forma permanente (ou, pelo menos, de longo prazo), com esclarecimentos de dúvidas mais comuns; também servem para publicar artigos
- correio eletrônico: forma de atendimento *offline* muito eficiente e que, apesar de mais antiga que as demais formas de comunicação entre alunos e professores, deve durar ainda por muito tempo

A terceira geração tem tanto cursos totalmente à distância quanto cursos semi-presenciais. Neste último, os encontros presenciais servem para tratar de dúvidas e servir para aulas práticas, em substituição aos – às vezes limitados – *chats* ou outra forma virtual de mediação.

Há ainda uma modalidade em que os alunos vão para uma sala de aula (nos chamados tele-centros) mas o professor encontra-se em outra localização, dando aula – normalmente ao vivo – e retirando dúvidas de vários alunos espalhados em salas de aula localizadas em regiões geográficas distintas. Essa modalidade funciona para aqueles alunos – normalmente adultos – que não conseguem assistir a aulas em vídeo em casa, seja por falta de disciplina, de infraestrutura ou de um ambiente propício à concentração.

Com o aumento de pesquisa e desenvolvimento no âmbito das TICs para uso em EAD, aumenta o número de opções tecnológicas a serem usadas nas aulas. A instituição de ensino e o professor precisam, portanto, observar as plataformas e escolher a(s) que mais se adequa(m) para cada propósito. O ponto em comum de todas as plataformas é a

possibilidade de interação e de promover experiências motivadoras aos alunos, o que ve se tornando inerente às metodologias de EAD da terceira geração.

A educação a distância teve seu primeiro registro na história em um anúncio de aulas de taquigrafia por correspondência ministradas por Caleb Phipps em 20 de março de 1728, na Gazeta de Boston, um jornal dos Estados Unidos. Na época, o professor enviava suas lições todas as semanas para os alunos inscritos.

Muito tempo depois, em 1833, na Suécia, uma universidade na cidade de Lund ofereceu um curso de composição por correspondência. Em 1840, na Inglaterra, o Skerry's College ofereceu cursos preparatórios para concursos públicos. Em 1856, na Alemanha, foram criados os primeiros cursos de idiomas a distância.

O século XIX foi caracterizado pelo surgimento de vários cursos a distância em todo o mundo, a exemplo dos pioneiros bem-sucedidos. A EAD ganhou espaço firme e se tornou importante e popular no contexto do ensino na segunda metade do século XX. Inicialmente, o objetivos dos cursos a distância era de aperfeiçoar e complementar formações já realizadas em cursos presenciais. Com o tempo, graduações completas puderam ser realizadas virtualmente.

Na Inglaterra, em 1969, surgiu a primeira universidade aberta, que consiste em uma instituição de ensino voltada exclusivamente à prática do ensino a distância. Este é considerado um marco na evolução da EAD, que mostrava ter fincado suas raízes na história da educação mundial. À época os alunos e professores comunicavam-se por meio materiais impressos de alta qualidade, já que havia agora uma universidade estruturada com foco em EAD. Logo em seguida, em 1972, foi criada a Universidade Nacional de Educação a Distância na Espanha.

### 1.3.2 EAD no Brasil

No Brasil, o EAD surgiu com cursos de especialização técnica profissional. O primeiro registro foi em 1904, com um anúncio nos classificados do Jornal do Brasil de um curso de datilografia em máquinas de escrever por correspondência.

Na década de 1920, o Brasil já contava com os primeiros cursos transmitidos pelas ondas do rádio, a novidade tecnológica da época. Os estudantes utilizavam material impresso para aprender Português, Francês e temas relacionados à radiodifusão.

Nas décadas de 1940 e 1950 surgiram cursos mais formais, sobre temas profissionalizantes, liderados pelo Instituto Rádio Monitor (criado em 1939), depois pelo Instituto Universal Brasileiro (criado em 1941) e pela Universidade do Ar (fundada em 1944), que teve sua segunda versão em 1947 patrocinada por algumas emissoras de televisão, pelo Senac e pelo Sesc. Até hoje algumas dessas instituições permanecem ligadas à formação profissional através de cursos a distância. Foi aí que surgiram os telecursos na televisão

em conjunto com material de apoio impresso enviado pelos Correios.

As décadas de 1960 e 1970 no Brasil foram marcadas por diversas iniciativas de EAD com o intuito de ampliar o acesso ao conhecimento, principalmente levando alfabetização e inclusão social a adultos. Com o passar do tempo, os cursos foram abrangendo demais níveis de ensino, como o fundamental completo para adultos. Cursos superiores via EAD no Brasil tiveram seu marco inicial no final da década de 1970 em Brasília.

Na década de 1970, a EAD passou a ser utilizada para capacitar professores por meio da Associação Brasileira de Tecnologia Educacional (ABT) e o Ministério da Educação (MEC), com os Seminários Brasileiros de Tecnologia Educacional.

Na década de 1970 também surgiram os telecursos na televisão em conjunto com material de apoio impresso enviado pelos Correios. Esses cursos conviviam com os formatos antigos de aulas através do rádio, modalidade que perdurou até meados da década de 1990, quando a publicação de materiais e as interações começaram a ser feitas através da internet.

Ainda sobre os telecursos, em 1978 foi criado o Telecurso 2<sup>a</sup> grau, uma parceria com apoio da Fundação Padre Anchieta e da Fundação Roberto Marinho. Seu foco era preparar jovens e adultos para exames supletivos de 2<sup>o</sup> grau. Em 1995, foi feita a sua reformulação, passando a se chamar Telecurso 2000, agora incluindo o curso técnico na área de mecânica com o objetivo de aperfeiçoar profissionais da área.

Foi nesse período que várias universidades formalizaram suas iniciativas EAD, até culminar com a criação, em 1996, da Secretaria de Educação a Distância (SEED), MEC. No mesmo ano a EAD no Brasil passou a contar com uma legislação abrangente que hoje garante, por exemplo, a validade de diplomas emitidos pelos cursos nesta modalidade. No Brasil, todo ano, vários novos cursos superiores são reconhecidos pelo MEC.

A Universidade de Brasília, uma das pioneiras na educação a distância, lançou cursos veiculados por jornais e revistas em 1979 e, dez anos depois, criou a plataforma CEAD (Centro de Educação a Distância) e lançou o BrasilEAD, um consórcio que nasceu de iniciativas públicas em prol da educação superior a distância no Brasil, com participação de outras instituições de ensino públicas estaduais e federais e do MEC.

Sobre o CEAD, é uma iniciativa bastante presente no contexto da EAD brasileira e tem apoiado o desenvolvimento de projetos e disciplinas de graduação que utilizem TICs para um ensino inovador. Para atingir tal objetivo, o centro elaborou o Guia de Atividades Educacionais com o uso de recursos tecnológicos de informação e comunicação a fim de orientar professores na montagem de aulas e atividades. Com isso, é possível encontrar na página da internet do CEAD sugestões de *softwares*, aplicativos e outras ferramentas a serem utilizadas na elaboração de aulas, sejam elas presenciais ou virtuais.

A EAD no Brasil teve dois outros grandes marcos: em 1999 foi criado o consórcio

CEDERJ (Centro de Educação a Distância do Estado do Rio de Janeiro) pela Fundação Centro de Ciências e Educação Superior a Distância do Estado do Rio de Janeiro – Fundação CECIERJ. O CEDERJ é formado por sete universidades públicas do Rio de Janeiro e um centro tecnológico. Em 2000 houve a criação da UNIREDE (Rede de Educação Superior à Distância), um consórcio que reúne 68 instituições públicas do Brasil. Após isso, em 2005, a EAD no ensino superior fincou de vez os pés em território brasileiro com a criação da Universidade Aberta do Brasil (UAB).

A UAB foi criada para ampliar a oferta de cursos de nível superior por meio da EAD e tem um papel social importante à medida que desenvolve um amplo sistema nacional de educação superior a distância. A prioridade é oferecer formação inicial a professores em efetivo exercício na educação básica pública, porém ainda sem graduação, além de formação continuada àqueles já graduados. Também há cursos na UAB destinados a gestores da educação na rede pública.

A UAB conta com polos de apoio para as atividades presenciais, como provas e laboratórios. Isso tem se tornado bastante comum nos cursos a distância promovidos por universidades públicas em todo o Brasil, como o CEDERJ. Os polos costumam ficar no interior do Estado, de forma a levar o ensino a lugares mais distantes das universidades. Com isso, os alunos têm um local físico para estudar caso necessitem de apoio, como bibliotecas e laboratórios de informática.

A Lei nº 9.394 – Lei de Diretrizes e Bases da Educação Nacional (LDB) – de 23 de dezembro de 1996 introduziu a educação a distância legalmente nas atividades de ensino no Brasil. Em seu artigo 80, a lei determina:

O Poder Público incentivará o desenvolvimento e a veiculação de programas de ensino a distância, em todos os níveis e modalidades de ensino, e de educação continuada.

Além da LDB, a EAD brasileira foi regulamentada pelos seguintes instrumentos:

- Decreto nº 5.622, de 19 de dezembro de 2005 (que revogou o Decreto 2.494, de 10 de fevereiro de 1998 e o Decreto nº 2.561, de 27 de abril de 1998)
- Portaria Ministerial nº 4.361, de 29 de dezembro de 2004 (que revogou a Portaria Ministerial nº 301, de 7 de abril de 1998)
- Resolução nº 1 do Conselho Nacional de Educação, de 3 de abril de 2001

Em um país com dimensões continentais como o Brasil, a metodologia de educação a distância tem chance de perdurar com sucesso por ser fundamental para romper

fronteiras e aproximar docentes e discentes, bem como para promover inclusão social e oportunidade de crescimento para quem não tem acesso à educação presencial onde mora. As iniciativas de EAD (bem como metodologias de aulas com auxílio de TICs) crescem a cada ano no Brasil. É preciso lembrar, no entanto, que o acesso à educação por meio da EAD é possível com fornecimento de infraestrutura de rede adequada.

## 1.4 Redes Sociais

Atualmente, com a tecnologia e a modernização dos processos de comunicação, as redes sociais são cada vez mais utilizadas. O seu propósito inicial – o de conectar pessoas – tem sido expandido para uso comercial, empresarial, de *marketing*, educacional, entre outros. Em particular, os alunos de hoje estão inseridos nesse contexto por meio de *smartphones*, *tablets*, *notebooks*, etc. Nesta seção será abordado o uso de redes sociais no ensino (em especial da Matemática), seus prós e contras, bem como a argumentação presente em alguns artigos que tratam do tema.

O uso de redes sociais em sala de aula tem suas desvantagens. Algumas delas são: eventual indisponibilidade dos recursos (*smartphones*, *tablets* e computadores, por exemplo) tanto por parte dos alunos como dos professores ou da escola; falta de domínio da rede social; dificuldades de infraestrutura como internet; possibilidade de os alunos ficarem dispersos, dadas as outras funcionalidades das redes sociais; e risco à privacidade, com a tênue fronteira entre o relacionamento professor-aluno e o relacionamento entre amigos (virtuais ou não).

Em uma matéria do IEEE (<<http://theinstitute.ieee.org/ieee-roundup/members/achievements/does-social-media-hinder-or-enhance-learning>>), Andreas Schleicher, diretor da Organização para a Cooperação e Desenvolvimento Econômico (OCDE), mostrou-se preocupado com a disponibilidade exacerbada de tecnologia aos jovens, que pode levá-los a buscar soluções prontas às questões matemáticas, ao invés de desenvolverem por eles mesmos. De acordo com ele, isso faria desses jovens pessoas preguiçosas e interferiria no desenvolvimento de habilidades de mais alto nível, que requerem maior processamento cognitivo.

Em sala de aula, as redes sociais podem abrir o campo de possibilidades de aprendizado e de acesso à informação. Por meio delas, é possível explorar diferentes métodos de ensino, como vídeos, fóruns de discussão e dinâmicas de grupo. Além disso, os estudantes podem compartilhar suas dúvidas e soluções no formato que acharem melhor. Um exemplo disso é o *site* <<https://artofproblemsolving.com>> no qual os alunos compartilham problemas e soluções e se tornou uma fonte de referência mundial em problemas de Olimpíadas de Matemática. O site tem um grupo no Facebook que funciona da mesma forma, revelando ser a rede social uma ferramenta de grande atratividade aos jovens que,



se usada corretamente, contribui para o ensino da Matemática. O número de canais no Youtube com assuntos matemáticos também é uma métrica do sucesso deste tipo de uso da rede social.

No artigo “O uso das redes sociais no ensino de Matemática através da monitoria”, CARVALHO, 2014, o autor descreve um experimento feito com alunos do Ensino Médio em que alguns deles realizavam monitoria via Facebook e, como resultado, todos obtiveram notas maiores (uns por ajudarem os colegas e outros por aprenderem de maneira mais direta e fácil com colegas da mesma turma). Nesse experimento, também houve aumento do interesse pela Matemática e os alunos tiveram maior interação, contribuindo no campo social.

No artigo “Redes Sociais na Educação: aprendizagem colaborativa no ensino de Matemática”, COSTA; FERREIRA, 2012, os autores utilizaram o Twitter como ferramenta no processo de construção colaborativa de conhecimento na Matemática. O parecer dos alunos foi positivo, ressaltando que o Facebook ou o Orkut poderiam ser melhores por apresentarem maior interatividade. O artigo deixa como ponto futuro a pesquisa sobre qual a melhor rede social a se utilizar neste caso e conclui que o desenvolvimento dos alunos foi melhorado com o uso de rede social.

No meio acadêmico internacional o assunto também é discutido e dois artigos interessantes defendem o uso de redes sociais no ensino da Matemática. Um deles é o “*Social Media and the Teaching of Mathematics in a Lifelong Learning Environment*”, NG; LATIF, 2011, que utilizou na *Open University Malaysia* as redes sociais Facebook e blog como ferramentas para ensinar Matemática, matéria que é considerada por muitos alunos adultos um pesadelo devido a experiências ruins no passado. Com interação – via rede social – entre alunos e tutores para assistência, orientação e suporte, foi criado um ambiente saudável de aprendizagem e o resultado foi positivo, reduzindo o isolamento de alguns estudantes e aumentando o aproveitamento da matéria.

O outro artigo é o “*Mathematics Teachers’ Perception of Using Social Media in Their Teaching in Tabuk, Saudi Arabia*”, ALBALAWI, 2017. Note-se que este é mais recente que os outros, demonstrando a tendência atual de se estudar a metodologia de uso de redes sociais no ensino da Matemática. Este artigo consistiu de uma pesquisa entre 142 professores de Matemática (82 homens e 60 mulheres) em diferentes escolas na cidade de Tabuk, na Arábia Saudita. Os resultados mostram que os professores avaliam como positivo o uso de redes sociais e a ferramenta mais utilizada foi o WhatsApp, com professoras mais favoráveis ao uso de rede social que professores do sexo masculino. No entanto, o artigo aponta como ponto a ser melhorado a inclusão da rede social na metodologia de ensino, visto que ela é mais usada sem uma meta específica e apenas como auxiliar em alguns trabalhos.

Pode-se concluir que há espaço a se explorar nas redes sociais para incrementar o

ensino da Matemática e este uso pode ser de grande valia, pois pode motivar os estudantes a aprender, produzir resultados melhores nos seus desempenho e aprendizado, além de diminuir a distância entre aluno e professor, assim como entre os alunos de uma turma.

Outra vantagem é que o recurso pode ser usado tanto com alunos de Ensino Médio como com alunos de graduação ou pós-graduação. Claro que também há riscos no uso de redes sociais no ensino, como a dispersão dos alunos e o imediatismo em procurarem respostas prontas ao invés de desenvolvê-las.

Para ser usada como ferramenta de ensino, a rede social, portanto, deve fazer parte do planejamento de ensino e da estratégia pedagógica, com plano de mitigação dos possíveis riscos que podem surgir com a nova metodologia.

## 1.5 Aplicativos

Existem diversos aplicativos utilizados no apoio do ensino da Matemática. A maioria deles funciona em sistemas operacionais *mobile*, mas também há aqueles com versões para computadores. A depender da idade dos alunos, o professor pode escolher um ou outro aplicativo. Para escolas com infraestrutura de laboratório de informática – ou ainda aquelas em que os alunos possuem computadores em sala – é comum o uso de aplicativos para computador.

Devido ao fato de a internet causar dispersão nos alunos em sala, muitas vezes opta-se por aplicativos e ferramentas que dispensem o seu uso, ou seja, os chamados aplicativos *offline*. Esses *softwares* podem exemplificar conteúdos ensinados em sala. No caso da Matemática, é comum o uso do GeoGebra no ensino da Geometria, por exemplo. Para crianças, o aplicativo Rei da Matemática desperta o seu interesse para fazer contas e resolver equações, desmistificando a Matemática e tornando a matéria divertida.

Nesta seção serão brevemente explicados os aplicativos GeoGebra e Rei da Matemática, com suas vantagens e seus desafios. Vale ressaltar que o primeiro tem um público alvo maior, já que a Geometria está presente em todo o Ensino Fundamental e Médio. Já o segundo é voltado a crianças das primeiras séries do Ensino Fundamental.

### 1.5.1 GeoGebra

Na Matemática, uma das áreas do Ensino Básico que demandam visualizações e desenhos é a Geometria. Os professores desenhavam com uso de régua, esquadro e compasso no quadro, porém, para a Geometria Espacial, a visualização torna-se mais dependente ainda das habilidades do docente em desenhar em perspectiva e de forma que os alunos consigam compreender. A Geometria Analítica, com as cônicas e seus elementos também dependem de habilidades de desenho por parte do professor. No estudo das funções (desde

o Ensino Fundamental com equações até o Superior com o Cálculo Diferencial e Integral), esboçar gráficos de funções complexas e encontrar suas raízes pode ser um desafio.

O *software* GeoGebra (junção das palavras **Geometria** e **Álgebra**) é um programa de Matemática Dinâmica que junta Geometria, Álgebra e Cálculo. Foi criado em 2001 pelo austríaco Markus Hohenwarter, gratuito, e é utilizado desde os níveis mais básicos de ensino da Matemática até o nível superior. O programa pode ser encontrado na página oficial <<https://www.geogebra.org/>>.

O GeoGebra é amplamente utilizado em aulas de Geometria (e não somente nelas, como será descrito nesta subseção), principalmente por facilitá-las com os desenhos e permitir observações mais complexas nos desenhos, o que seria feito de forma limitada com desenhos em quadro. Uma das principais vantagens do programa é a possibilidade de fixar alguns elementos do desenho e mover outros, auxiliando, por exemplo, na exibição de ângulos, medidas ou propriedades invariantes. Por essa principal característica de um único desenho se transformar em várias outras opções sem perder suas propriedades geométricas é que o GeoGebra é considerado um aplicativo de Geometria Dinâmica.

De acordo com SIQUEIRA; CAETANO, 2016, um *software* de Geometria Dinâmica é “um ambiente que permite simular construções geométricas no computador”. De fato, construções geométricas feitas com o auxílio do computador são mais interativas que aquelas tradicionais com régua e compasso. No entanto, objetos feitos pelas primeiras continuam preservando todas as propriedades daqueles construídos pelas últimas.

Com isso, é possível deduzir as propriedades das figuras geométricas e trabalhar os conceitos de reflexão, translação e rotação, congruência, semelhança e cálculo de ângulos.

Por meio da movimentação das figuras em diversas direções, é possível mudar algum parâmetro, comparar e voltar ao aspecto inicial. Para provar que a soma dos ângulos internos de um triângulo é 180, por exemplo, é possível alterar a figura e ver empiricamente que, quando um dos ângulos aumenta, os outros diminuem, mas a soma é sempre 180.

SAHA; AYUB; TARMIZI, 2010, em estudo realizado pela *Universiti Putra Malaysia* sobre os efeitos da utilização do GeoGebra no ensino de coordenada geométricas mostrou que as aulas que contaram com auxílio do aplicativo como complemento às aulas realizadas em sala são mais significativas do que somente aulas tradicionais em que os alunos são meros ouvintes no processo de ensino.

SILVA; SANTIBANEZ; ZUASNABAR, 2018 fizeram uma experiência em sala de aula com alunos do 3º ano do Ensino Médio no campo da Geometria Analítica; como resultado, obtiveram êxito quanto ao desempenho e à boa aceitação do GeoGebra pelos alunos. Também com alunos do 3º ano do Ensino Médio, SILVA, 2016 relatou que o *software* contribuiu para a aprendizagem dos conteúdos de ponto e reta.

A pesquisa de PEREIRA, 2012 também concluiu que o programa contribuiu para a consolidação do conhecimento de triângulos, bissetrizes e circunferências. Após o uso do GeoGebra em sala com alunos do Ensino Médio, constatou:

“Os alunos tiveram a oportunidade de validar suas hipóteses, conjecturar sobre possíveis caminhos para a solução das tarefas e discutir de forma colaborativa suas soluções encontradas.”

Não se pode esquecer, porém, da outra grande utilidade do programa que serve de auxílio no ensino da Matemática: o estudo das funções, com a possibilidade de desenhar seus gráficos e analisá-las em termos de domínio, imagem, raízes, extremos, concavidades, crescimento, etc.

O estudo realizado por MACEDO et al., 2017 mostrou que o uso do GeoGebra trouxe benefícios ao aprendizado de função afim por alunos do 1º ano do Ensino Médio, despertando seu interesse pelo conteúdo e facilitando o entendimento por meio da visualização de gráficos no aplicativo. No estudo, 96% dos alunos afirmou que o uso de recursos tecnológicos torna mais fácil a compreensão do conteúdo função afim, assim como 100% deles achou a aula mais atrativa com o uso do aplicativo e apenas 4% deles encontrou dificuldade em manusear o GeoGebra. Outro estudo também aplicado por SIQUEIRA; CAETANO, 2016 a alunos da mesma série no âmbito do estudo de funções obteve resultados similares.

Cada vez mais os alunos também vêm usando o programa para auxiliar na resolução de questões. O GeoGebra vem chamando a atenção de pesquisadores e tem sido tema de diversas investigações didáticas. Além de contar com muitos recursos, ele é gratuito, simples de ser usado e possui um tutorial bastante útil e explicativo.

### 1.5.2 Rei da Matemática

O aplicativo Rei da Matemática (ou *King of Math*), totalmente em português, serve para crianças a partir de 8 anos de idade e é uma espécie de jogo disponível tanto para iOS como para Android. A interface mostra um Rei dentro de um reino que terá que acertar cálculos para aniquilar os seus inimigos. Há uma versão gratuita com os módulos de adição, subtração e um módulo chamado “combinado 1”. A versão paga também inclui os módulos de multiplicação, divisão, aritmética, geometria, frações, potências, estatística, equações e um módulo chamado “combinado 2”.

Em sala de aula, pode-se usar o aplicativo como forma de estimular o interesse dos alunos nas contas, principalmente nos níveis mais avançados. Pode-se tomar alguns exemplos no aplicativo do professor (ele projetaria a imagem de seu aplicativo no quadro

ou, caso não possuísse infraestrutura necessária, copiaria o desafio no quadro), para que a turma participe em conjunto.

Após a turma demonstrar entendimento no funcionamento e nas regras do jogo, o professor separa a turma em dois times com base em sorteio e reproduz no projetor (ou escreve no quadro) os desafios daquele nível que o jogo lançou, com o cronômetro de tempo que o aplicativo já possui. O time que responder primeiro ganha um ponto nas questões de nível 1, dois pontos nas de nível 2, etc. O professor anota no quadro a pontuação em tempo real dos times. Também é possível, caso todos tenham celular/*tablet*, que cada um faça no seu e o professor veja o score posteriormente.

Esta dinâmica pode continuar por mais ou menos tempo dependendo do tempo de aula. O time vencedor pode ser premiado com itens materiais, como bala, chocolate, canetas corporativas se a escola possuir, ou mesmo com um valor pequeno na pontuação final do bimestre (por exemplo, 1 décimo na média do bimestre). Isso serve como incentivo para os alunos participarem de forma mais ativa. É possível, também, fazer a competição individual se for o caso de estimular a competitividade, mas é preciso que o professor deixe claro o fato de ser mais importante competir do que ganhar.

Com o uso deste aplicativo, o aluno desenvolve sua capacidade de resolver contas com maior velocidade e pensar de forma não convencional, como nos problemas em que é dada a resposta e pede-se a conta que gera aquele número. Com isso, desenvolve-se no aluno a capacidade de montar e resolver equações simples de primeiro grau mesmo no 2º ano de forma lúdica (equações do tipo “ $? - 12 = 2$ ”). Vale ressaltar que neste plano de aula a ideia é usar a versão gratuita, pois adição e subtração fazem parte da ementa do 2º ano do Ensino Fundamental.

Existe, no entanto, a possibilidade de dificuldades no uso do Rei da Matemática em sala de aula: dificuldades técnicas, falta de infraestrutura necessária, como falta de projetor para que os alunos possam compartilhar a tela do aplicativo do professor na primeira parte da aula, ou mesmo falta de disponibilidade de os alunos terem seu próprio aplicativo (dificilmente eles possuirão celular/*tablet* nesta série).

Outra dificuldade vem do fato de o aplicativo estar instalado no celular ou *tablet* e os alunos podem ficar dispersos. Este risco é mitigado quando o professor usa somente o aplicativo dele e projeta na tela para todos. Uma vantagem deste aplicativo quando comparado com outros é o fato de rodar sem necessidade de internet, o que poderia ser um risco à aula.

## 1.6 Prós e Contras

A literatura é vasta e elenca alguns prós e contras mais comuns do uso de tecnologias na educação.

Nos dias de hoje, quadros brancos em sala de aula são comuns. Isso é resultado do avanço tecnológico, já que, ano atrás, as salas de aulas tinham apenas quadros de giz (hoje considerados por muitos como antigos).

Também não é raro que muitas escolas utilizem recursos tecnológicos, como computadores e projetores (em substituição às já antigas transparências), o que representa outro exemplo de tecnologia incorporada ao ensino.

Uma aula com um programa de animação pode ilustrar uma ideia mais facilmente do que apenas apelar para a imaginação do estudante. A geração Z, nascida entre 1988 e 2010, é extremamente visual e oferecer essa forma como técnica de aprendizado é uma ótima opção. Mapas mentais feitos no computador também podem auxiliar os alunos a assimilarem conteúdos extensos.

Uma modalidade de ensino que tem crescido bastante é a Educação a Distância (EAD), já tratada neste capítulo. Ela é resultado direto da aplicação da tecnologia na educação e deve ser usada com cautela, principalmente no âmbito da Educação Básica, seja ela no contexto do ensino infantil, fundamental ou médio. A modalidade EAD é a que mais cresceu – e continua crescendo – no Brasil e no mundo nos últimos 10 anos. Aplicada principalmente a cursos de adultos, muitas vezes é confundida como sinônimo de tecnologia na educação.

Como discutido anteriormente, com a EAD é possível ampliar o acesso a conteúdos, cursos e treinamentos a mais alunos em diversas localizações geográficas. Sua principal vantagem é que o aluno não precisa comparecer fisicamente ao local da aula. Com isso, diminui o número de evasões por motivos de falta de disponibilidade de horário para os alunos comparecem às aulas presenciais, já que, na EAD, eles podem assistir às aulas no horário e no local mais conveniente.

Resumindo, alguns prós da tecnologia aplicada ao ensino são listados abaixo:

- aulas com recursos tecnológicos podem ser mais dinâmicas
- tecnologias levam a mais alunos por aula, o que reduz o custo da aula por aluno
- aulas que fazem uso da tecnologia tendem a despertar maior interesse dos alunos e captar sua atenção
- a tecnologia é aliada do professor à medida que poupa a sua voz com o uso de, por exemplo, microfone

- com o auxílio da tecnologia, é possível ter turmas maiores e levar aulas para muito mais alunos que não teriam oportunidade de assistir a determinada aula se não fosse a internet
- a tecnologia facilita o acesso a informações em caso de dúvidas
- o estudo individualizado com auxílio de tecnologias, como a EAD, permite ao aluno desenvolver disciplina de autoaprendizagem
- equipamentos de realidade virtual podem suprir a falta de recursos físicos nos laboratórios, além de possibilitarem imersão prática do conteúdo teórico passado em sala de aula
- fóruns de internet com o propósito de debater temas abordados em sala de aula podem fomentar a troca de informações e facilitar o aprendizado
- as tecnologias tendem a ser mais inclusivas, uma vez que telas de computadores, *tablets smartphones* são adaptáveis para pessoas com deficiências motoras, auditivas ou de visão

Os contras do uso da tecnologia em sala de aula também são diversos. Eles provêm principalmente de limitações de recursos computacionais das escolas, limitações de conhecimento pelos professores e alunos das ferramentas e o fato de ser aplicada de forma errada, comprometendo o aprendizado, a concentração ou a formação dos alunos.

Para que uma escola esteja preparada para receber os recursos tecnológicos como, por exemplo, *smarTV*, computadores, rápida internet sem fio, laboratórios de computação e projetores, é necessário um alto investimento que, nem sempre, é possível de ser feito.

Um outro ponto bastante discutido e consensual no uso de internet em sala de aula – seja ela acessada pelo celular ou pelo computador pelo aluno – é a distração que ela causa, se não for feita de forma organizada. O professor não consegue competir, por exemplo, com as redes sociais. Portanto, para o seu uso é preciso planejamento em Tecnologia da Informação para que a rede disponibilize apenas acessos condizentes com o propósito da aula.

Por isso, nas aulas, o docente deve redobrar sua atenção e ver se o estudante está realmente utilizando o recurso da forma proposta ou apenas para entretenimento – o que deve ser repreendido. Existem diversas sugestões de solução para essa questão. Conforme citado acima, é possível restringir o acesso a apenas páginas e sistemas que façam parte do contexto da aula. Outra solução é incentivar os alunos a fazerem a tarefa e não dispersarem por meio de, por exemplo, pontuações, brindes ou tempo livre para aqueles que cumprirem a atividade planejada e participarem com suas ideias e opiniões – tanto presencialmente em sala como em fóruns criados sobre a matéria.

Para que isso funcione de maneira correta, no entanto, o docente precisa estudar previamente a funcionalidade das ferramentas de aprendizagem que irá utilizar em sala, bem como prever possíveis problemas, com alternativas para esses casos.

O escritor Marc Prescky citou em um artigo que a falta de entrosamento do professor com as novas tecnologias é um dos grandes motivos pelos quais o objetivo da aula não é atingido. Ou seja, a aula programada para guiar os alunos no aprendizado de determinado conteúdo acaba tornando-se um momento de lazer, fazendo com que o professor desista de implantar as ferramentas em suas aulas e retornando ao modelo tradicional após uma experiência ruim nesse contexto.

Abaixo seguem alguns contras do uso da tecnologia na educação. O primeiro item tende a ser polêmico, uma vez que se pode interpretá-lo como uma vantagem das TICs no processo de ensino, à medida em que o professor fica obrigado a se reciclar, o que é bom, mas, sob o ponto de vista do esforço demandado por ele, é considerado por CONEXIA, como uma desvantagem.

- aulas com recursos computacionais tendem a requerer um planejamento de aula mais detalhado
- muitas instituições de ensino não dispõem de infraestrutura para levar tecnologia para a sala de aula
- muitos docentes não têm conhecimento de uso das tecnologias e, para usá-las como ferramenta de ensino, é preciso treinamento, nem sempre possível
- a depender de como se utiliza a tecnologia, ela pode limitar a criatividade e a construção do raciocínio (principalmente em crianças), tendo em vista que se facilita o acesso à informação sem, às vezes, dar tempo para que se construa deduções que guiem os alunos ao entendimento completo do conteúdo
- os dispositivos eletrônicos, se usados demasiadamente, causam problemas de visão e podem reduzir concentração e atenção, além de falta de sono e risco de ansiedade e depressão
- a modalidade de educação a distância ou mesmo o uso individualizado de *tablets* e *notebooks* em sala de aula diminuem o contato social entre as crianças, importante para a sua formação



## 2 A linguagem de programação R

Neste capítulo será descrita a linguagem de programação R, escolhida para ser trabalhada de forma lúdica com os alunos do Ensino Médio e na qual será implementado o exercício ilustrativo da técnica de *inpainting*.

### 2.1 A Linguagem

A linguagem R foi criada a partir da sua antecessora S que foi criada na década de 70.

Por volta dos anos 90, devido à necessidade de uma linguagem de código aberto (ou *open source*) para cálculos estatísticos e análise de dados, foi criado o R, uma linguagem utilizada por cientistas, estatísticos e, mais recentemente, cientistas de dados como um meio conveniente para a análise exploratória de dados interativos.

Ao contrário do que se imagina sobre a linguagem R, ela não é limitada a apenas sessões iterativas, pois devido ao fato de ser uma linguagem de programação, os *scripts* podem ser criados e empacotados como bibliotecas. As soluções com base em *scripts* fornecem resultados mais consistentes, rápidos e confiáveis do que os fluxos de trabalho tradicionais, que requerem uma grande quantidade de interações manuais com uma interface gráfica de usuário. Mas deve-se lembrar que é possível encapsular um código em R num executável para ser utilizado diretamente ou mesmo em outro sistema desenvolvido em C ou Python, por exemplo.

Muitos consideram o R não somente uma linguagem, mas um programa ou *software*, assim como o Matlab e o SAS. Este é um tema amplamente discutido quando se considera a manipulação de dados usando planilhas do Excel ou um banco de dados relacional como o servidor SQL. A linguagem R é mais usada para manipulação de conjuntos de dados de tamanho médio, análises estatísticas e produção de documentos, gráficos, relatórios e apresentações centradas em dados.

Além disso, o R fornece uma ampla variedade de modelagens lineares e não lineares, testes estatísticos clássicos, análise de séries temporais, *clustering*, além de ser altamente extensível. Um dos pontos fortes do R é a facilidade aliada à qualidade para desenhar rapidamente gráficos, histogramas, incluir símbolos matemáticos e fórmulas, quando necessárias, já que o R é um conjunto integrado de facilidades de *software* voltado para manipulação de dados e exibição gráfica.

Como exemplo de onde se utiliza o R, duas das maiores empresas presentes no mercado, Facebook e FourSquare, utilizam a linguagem R tanto para recomendações quanto

para modelagem de comportamentos de usuários.

A linguagem R é uma linguagem de programação dinamicamente tipada, ou seja, é possível modificar os tipos de dados de variáveis em programas que já estejam em execução; com isso, o R provê uma melhoria na hora de programar, pois não se precisa realizar conversão dos tipos de dados. Por outro lado, isso requer mais cuidado do programador, já que, com a tipagem livre, podem-se igualar *arrays* ou matrizes de dimensões distintas, por exemplo.

É muito comum usar o RStudio (compilador do R) para programar em R. O RStudio também pode ser utilizado para criar documentos HTML ou mesmo arquivos .pdf e apresentar *templates* HTML. Ele simplifica processos comuns como a exportação de imagens e importação ou exportação de arquivos de dados. Ele fornece visibilidade para uma grande quantidade de informações sobre o ambiente e os comandos executados do R de forma bastante interativa durante uma sessão.

Dessa forma, o RStudio possibilita um aumento ainda mais expressivo do R como um ambiente interativo para análises estatísticas e gráficas, além de também aumentar a capacidade do R para apresentar, de forma mais simples, os conceitos complexos usando expressões simples ou mesmo interações.

Hoje, a linguagem R conta com mais de 5000 pacotes disponíveis para utilização de forma gratuita no CRAN (<https://cran.r-project.org/>), além de vários outros pacotes disponibilizados através do GitHub (<https://github.com/>). Estes pacotes geralmente oferecem APIs simplificadas que funcionam como linguagens de domínio específicas para especialistas que trabalham em determinadas áreas.

Devido à flexibilidade do R e à natureza *open source* da linguagem, podem ser encontrados pacotes inconsistentes e mal estruturados, portanto, deve-se ficar atento a esse aspecto. Grande parte dos pacotes disponibilizados, porém, estão se tornando mais consistentes em termos de estrutura e função. Um dos criadores, Hadley Wickham, disponibilizou uma série de pacotes confiáveis relacionados à manipulação de dados, plotagem e criação de pacotes para a linguagem R. A biblioteca ggplot2 citada anteriormente neste trabalho é um bom exemplo da qualidade oferecida nas bibliotecas.

A linguagem R é uma linguagem poderosa e propícia a ambientes estatísticos que estão realizando incursões em locais nos quais o processamento de dados se torna cada vez mais sofisticado. Com o uso das bibliotecas disponíveis – tomando-se o devido cuidado para escolher aquelas mais confiáveis e utilizadas – é fácil realizar tarefas complexas com o uso de apenas uma ou duas funções, de forma a suprir necessidades matemáticas e processamentos de dados mais rebuscados.

Contudo, o R continuará crescendo no quesito popularidade entre os profissionais de Tecnologia da Informação (TI), tendo em vista a necessidade crescente de produção

de informações e análise de dados. Tal análise se torna cada vez mais importante em um mundo que gera mais dados a serem processados. Além disso, exige-se tempestividade e confiabilidade nos gráficos e relatórios produzidos na área de TI.

## 2.2 Utilizando o R

Para melhor usabilidade, recomenda-se o uso do RStudio, que é a IDE, isto é, a interface gráfica, utilizada para a linguagem R. O RStudio é o melhor ambiente disponível para desenvolvimento de códigos em R. Para realizar o *download* basta acessar o site oficial: <https://rstudio.com/products/rstudio/download/>.

Em busca de um melhor aproveitamento em relação ao entendimento do funcionamento do R, este capítulo abordará algumas de suas funções básicas, assim como apresentará alguns de seus pacotes (também chamados de bibliotecas) mais utilizados.

Assim que instalado, o R já vem com uma grande quantidade de pacotes pré-instalados e exemplos de códigos. Muitas vezes é necessário, no entanto, adicionar pacotes mais específicos para determinada tarefa.

Após a instalação dos pacotes que se fizerem necessários, é preciso carregá-los no R (o mais fácil é fazê-lo a partir da interface do RStudio, mas, por ora, é apresentada a forma via linha de comando). Ao carregá-los, caso estes tenham dependências, estas serão carregadas automaticamente, sem necessidade de instalação prévia.

Na linha de comando, para carregar um pacote instalado o comando utilizado é o seguinte:

```
library(<<nome do pacote>>)
```

Até o momento foram apresentados os comandos a serem executados na linha de comando do R, que é a distribuição padrão instalada com o programa. Com a ajuda do RStudio, o desenvolvimento permite mais interação, tendo em vista que ele oferece teclas de atalho, conclusões de código, gerenciamento de janelas e interações gráficas que podem ser utilizadas ao invés das chamadas de funções complexas por meio da linha de comando.

Os exemplos e os códigos aqui gerados serão feitos com o apoio do RStudio.

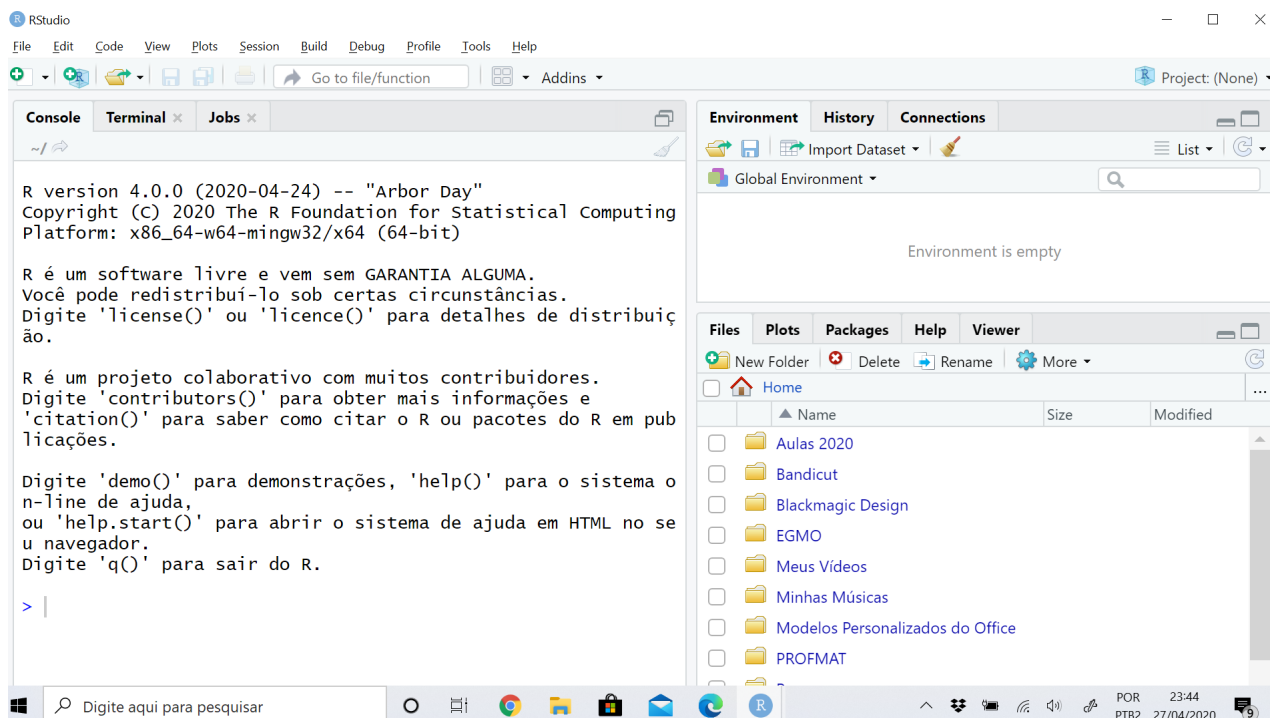


Figura 2.1 – Tela Inicial do RStudio

Primeiramente, é importante ressaltar que a comunidade do R é bastante ativa, o que facilita a retirada de dúvidas sobre a sua utilização. Existem diversos *sites* com informações e exemplos de códigos em R. Um deles é o <https://csgillespie.github.io/efficientR/setup.html#rstudio>. Além disso, é útil consultar a documentação oficial da linguagem, principalmente por meio do comando **help()**, com a seguinte sintaxe:

$$\text{help}(\langle\langle\text{nome da função}\rangle\rangle)$$

O comando **help()** ajuda o usuário a ter um breve contato com a função que está pesquisando e traz como principais as seções:

- **Description:** descreve a função
- **Usage:** mostra como é feita a chamada da função
- **Arguments:** exhibe os parâmetros de entrada da função
- **Value:** descreve o retorno da função
- **References:** traz a bibliografia onde se pode obter mais detalhes da função
- **See Also:** lista demais funções que podem ser úteis na busca daquele contexto
- **Examples:** exhibe exemplos de uso da função

Voltando à tela do RStudio da Figura 2.1, o lado esquerdo contém as duas abas principais: **Console** e **Terminal**. Na primeira é possível rodar os códigos em R e executar os comandos referidos como “linha de comando” neste trabalho anteriormente. A segunda aba é um terminal como o do computador, aquele que se inicia com o código **cmd** no menu Iniciar (no caso do Windows).

Além de executar comandos linha a linha no Console do RStudio, é possível criar um arquivo de extensão **.r** e executá-lo. O código no arquivo de extensão **.r** pode ser desde uma simples sequência de comandos que poderiam ser executados um a um no Console, como também um código mais rebuscado que inclui definição de variáveis e funções, por exemplo.

Para criar um arquivo **.r** no RStudio, basta selecionar o menu **File** → **New File** → **R Script** ou por meio do atalho Ctrl+Shift+N. O R executará as linhas na ordem do arquivo.

Para fins de exemplificação, segue abaixo código que usa a biblioteca de construção de gráficos estatísticos **ggplot2**. Esse código foi salvo como “exemploR.r” e está nos anexos deste trabalho.

Antes de executar os comandos que fazem uso dela, é preciso instalá-la por meio do comando **install.packages(“ggplot2”)**. O R traz algumas pequenas bases de dados para fins de exercício. Aqui será usado o **mtcars**.

No exemplo em questão, é usada a função **ggplot** com os dados da base **mtcars** conforme mencionado anteriormente. O segundo parâmetro diz respeito à visualização do gráfico, que tem a coluna **disp** no eixo das abscissas e a coluna **mpg** no eixo das ordenadas, bem como define as cores no gráfico, a depender dos valores da coluna **am**. Por fim, é definido o ponto geométrico para identificar cada ponto do gráfico de dispersão.

```
1 library (ggplot2)
  ggplot(data = mtcars, aes(x = disp, y = mpg, colour = as.factor(am))) +
3   geom_point()
```

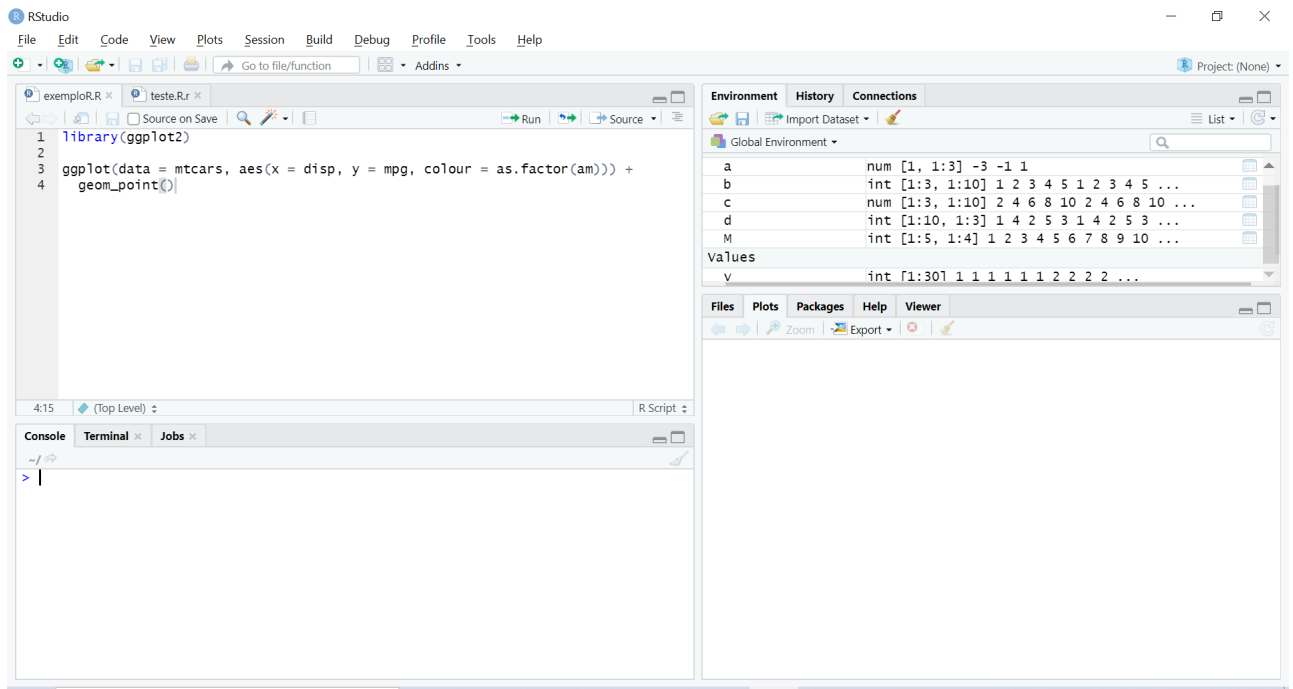
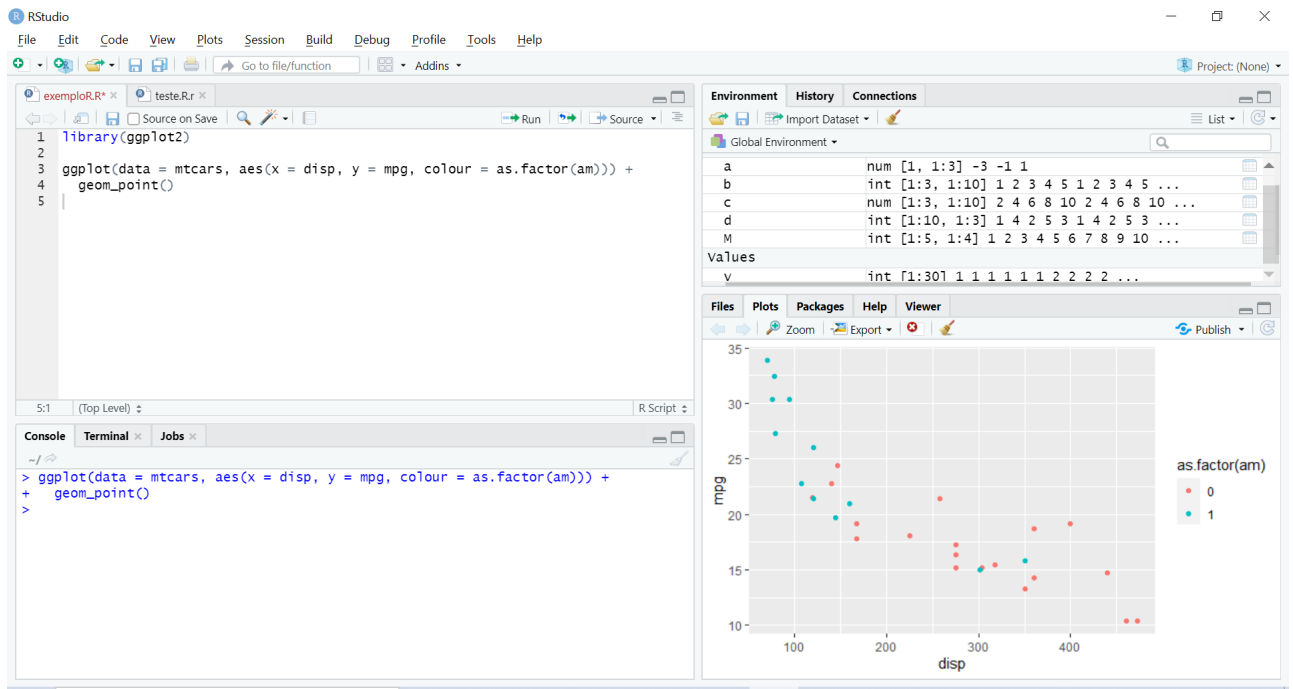
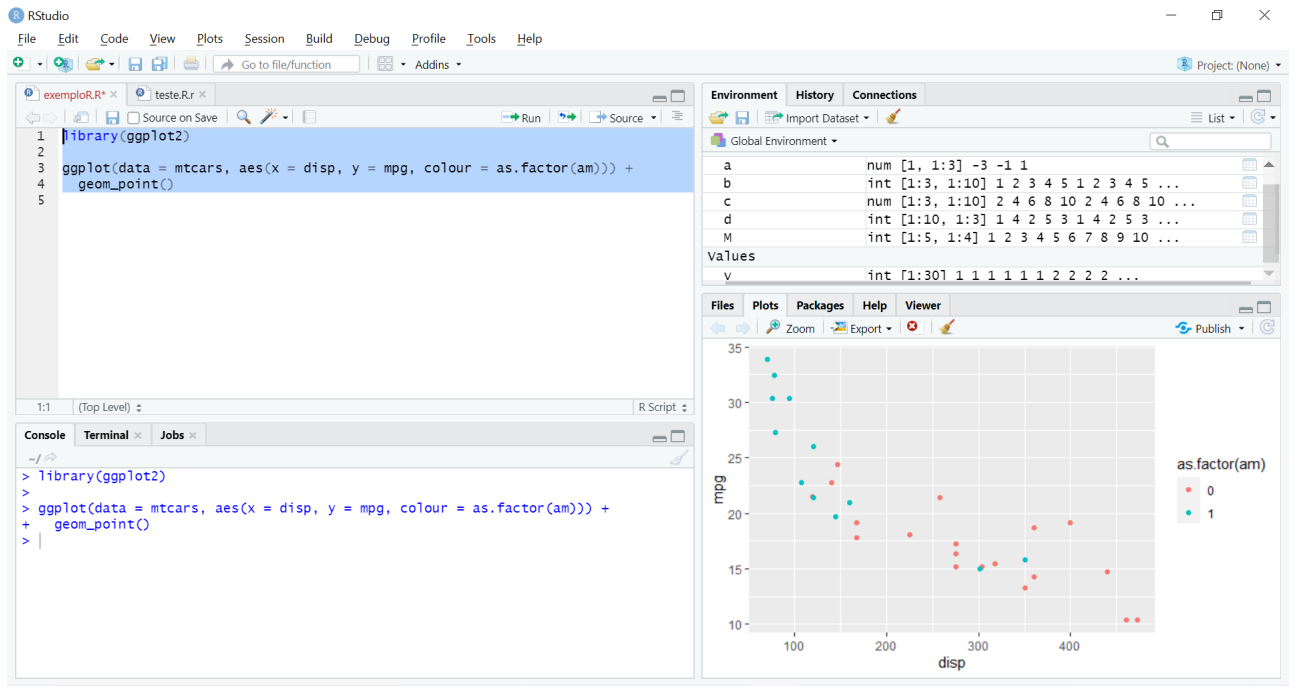


Figura 2.2 – ExemploR.r em R

Para a execução do código que está no arquivo `.r`, há duas opções: a primeira é por meio do menu **Run** que fica no segundo quadrante no RStudio, no canto superior direito logo acima do arquivo com o código-fonte. A segunda maneira é selecionar tudo e utilizar o atalho `Ctrl + Enter`. Esse atalho serve para executar algumas partes do código, basta selecionar o que se deseja executar e utilizá-lo.

Em ambos os casos, o gráfico gerado fica no quarto quadrante no RStudio.

Figura 2.3 – Execução do ExemploR.r pelo menu **Run**Figura 2.4 – Execução do ExemploR.r pelo atalho **Ctrl + Enter**

## 2.2.1 Operações Básicas

As operações aritméticas são feitas de forma intuitiva, como se fosse uma calculadora. Ou seja, se for digitado o comando  $5 \wedge 2 + 3 * 10 / 5$ , a resposta é 31. O operador

módulo é o menos trivial, denotado por `%%`. Por exemplo, a resposta da operação `18 %% 7` é 4.

As operações lógicas também são as conhecidas, com retorno do tipo *booleano*, ou seja, que assume os valores TRUE ou FALSE. Por exemplo, o retorno do comando `5 > 5` é FALSE, enquanto o retorno de `5 >= 5` é TRUE. Uma observação acerca dos valores verdadeiro e false é que, no R, esses valores precisam ser escritos com todas as letras maiúsculas: TRUE e FALSE.

O teste para saber se um valor é igual ao outro é feito com dois símbolos de igualdade, como no comando `3 == 3`, que retorna TRUE. Já o teste de valores diferentes é feito com `!=`, como no comando `3 != 3`, retornando FALSE. Além disso, os operadores lógicos E e OU são denotados por `&` e `|` respectivamente.

No R o valor *booleano* TRUE é visto como 1, enquanto FALSE vale 0. Ou seja, é possível executar o comando `TRUE + TRUE`, que dá o número 2 como resposta, ou o comando `TRUE/4`, que retorna 0,25.

Para identificar o tipo de uma variável ou valor, basta utilizar a função `class()` conforme sintaxe a seguir:

```
class(<<nome da variável>>)
```

Como exemplo, o comando `class(3)` retorna **numeric**, já `class(FALSE)` retorna **logical**.

### 2.2.2 Atribuição de Valores a Variáveis

A linguagem R é dinamicamente tipada, o que quer dizer que o tipo de uma variável é atribuído quando se dá um valor a ela.

Para atribuir um valor a uma variável (ou o valor de uma variável a outra), usa-se o símbolo de igualdade, setas ou a função `assign()`, conforme ilustrado abaixo comandos que atribuem o valor 4 à variável x.

```
x = 4
```

```
x <- 4
```

```
4 -> x
```

```
assign("x",4)
```

Assim, para criar uma variável, basta atribuir um valor a ela, conforme exemplo a seguir, que retorna o valor 10 quando executadas as duas linhas na ordem:



```
x = 4
```

```
x + 6
```

Um erro comum que deve ser monitorado é usar apenas um símbolo `=` para testar se uma variável é igual a outra. Isso fará a primeira receber o valor da segunda, ao invés de retornar o valor `TRUE` ou `FALSE`, já que, para esse propósito, o correto é usar dois símbolos: `==`.

### 2.2.3 Vetores

Até então foi visto como armazenar dados em variáveis. No entanto, algumas vezes é necessário armazenar uma sequência de dados. Por exemplo, pode-se desejar armazenar as idades de uma turma de 30 alunos. Criar uma variável para cada um dos alunos não parece ser melhor forma de fazê-lo. É para isso que o R dispõe de alguns objetos capazes de lidar com grandes quantidades de dados de uma maneira mais inteligente.

Um desses objetos chama-se vetor. Na linguagem R, um vetor é uma sequência de dados de um mesmo tipo. Para criar um vetor na linguagem R, a função utilizada é a `c` e o processo é bastante simples. Como mencionado anteriormente, só é possível armazenar dados do mesmo tipo num vetor.

A seguir é exibido um exemplo de criação de três vetores: um com os componentes `{-3, 11, 29}` atribuído a uma variável `y`, outro composto dos textos `{"Inpainting", "Programação"}` atribuído à variável `k` e um terceiro vetor `v` com os componentes *booleanos* `{TRUE, FALSE, TRUE}`.

```
y <- c(-3, 11, 29)
```

```
k <- c("Inpainting", "Programação")
```

```
v <- c(TRUE, FALSE, TRUE)
```

É permitido, em R, fazer algumas operações numéricas com vetores cujos componentes são números. No exemplo anterior, no qual `y` é o vetor `(-3, 11, 29)`, é possível fazer a multiplicação desse vetor por um escalar utilizando a notação `2*y`, retornando o vetor `(-6, 22, 58)`, ou seja, todos os componentes são multiplicados pelo número.

$$2 * \begin{pmatrix} -3 \\ 11 \\ 29 \end{pmatrix} = \begin{pmatrix} -6 \\ 22 \\ 58 \end{pmatrix}$$

Também é possível dividir o vetor  $y$  por algum escalar e o R divide todos os componentes pelo número em questão. De forma análoga às operações com números, é possível somar e subtrair dois ou mais vetores.

Outra utilidade no objeto vetor é a possibilidade de acessar apenas uma determinada posição  $i$ , com  $1 \leq i \leq n$ , onde  $n$  é o tamanho do vetor, ou seja, o número de componentes. Isso é feito com colchetes. Por exemplo, para acessar o sexto elemento do vetor  $\mathbf{v}$ , usa-se o comando  $\mathbf{v}[6]$ .

Existem diversas funções envolvendo vetores. As mais importantes estão listadas a seguir, com os respectivos retornos. Todas as funções recebem como parâmetro um vetor, exceto a que calcula correlação, que recebe dois vetores como entrada.

- **sum()**: soma dos elementos do vetor
- **mean()**: média dos elementos do vetor
- **sd()**: desvio-padrão dos elementos do vetor
- **median()**: mediana dos elementos do vetor
- **var()**: variância dos elementos do vetor
- **cor()**: correlação entre dois vetores
- **min()**: valor mínimo entre os elementos do vetor
- **max()**: valor máximo entre os elementos do vetor
- **range()**: valores mínimo e máximo entre os elementos do vetor
- **summary()**: algumas estatísticas dos elementos do vetor, como valor mínimo, média, mediana, quartis, etc.
- **sort()**: vetor ordenado em ordem crescente
- **length()**: comprimento do vetor, ou seja, número de elementos
- **is.na()**: para cada posição do vetor, retorna TRUE ou FALSE, sendo o último o caso de a posição correspondente não estar disponível. Isso ocorre quando há um erro e o R atribui o valor **NA**

## 2.2.4 Matrizes

O conceito de matriz no R é similar ao conceito de matriz definido na Matemática. Seus componentes são identificados pelos índices de linha e coluna correspondentes. Para

o R, uma matriz é uma coleção de elementos de uma mesma classe arranjados em duas dimensões.

O comando para definir matriz no R utiliza a função **matrix()**, que possui três parâmetros de entrada obrigatórios: os dados que comporão a matriz, o número de linhas e o número de colunas, nesta ordem.

Ao listar os dados para o preenchimento da matriz, o padrão do R é fazer o preenchimento por colunas – e não por linhas. No entanto, é possível mudar isso incluindo um quarto parâmetro de entrada chamado **byrow = TRUE**. Os dados precisam ter um tamanho que seja divisor do número total de elementos da matriz ( $\#linhas * \#colunas$ ). Dessa forma, caso o tamanho do primeiro parâmetro de entrada não seja igual ao total de elementos, o preenchimento repete os dados de entrada – sempre por colunas – até o completo preenchimento.

Existem algumas maneiras de listar os dados de entrada da matriz. Uma delas é o vetor, já abordado neste capítulo. Outra é usar a função **seq()**, que lista os números inteiros em sequência desde um número inicial até um número final, inclusive. A sintaxe é uma das definidas abaixo, separando os dois números por vírgula ou dois pontos, sendo que a segunda forma não é válida se a chamada da função incluir outro argumento, como o **byrow**, por exemplo.

```
seq(<<número_inicial, número_final>>)
```

```
seq(<<número_inicial: número_final>>)
```

É possível definir o incremento dessa sequência por meio de um terceiro argumento. O padrão de incremento é 1. Abaixo seguem alguns exemplos de uso da função **seq()** com os respectivos retornos:

```
seq(1:5): 1 2 3 4 5
```

```
seq(1,5): 1 2 3 4 5
```

```
seq(3,11,2): 3 5 7 9 11
```

```
seq(3,11,3): 3 6 9
```

A definição de matriz no R por meio da função **matrix()** é feita conforme os exemplos abaixo:

```
a <- matrix(c(-1,0,-2,-1),2,2):  $\begin{pmatrix} -1 & -2 \\ 0 & -1 \end{pmatrix}$ 
```

$$\mathbf{a} \leftarrow \mathbf{matrix}(\mathbf{seq}(-2,3),3,2): \begin{pmatrix} -2 & 1 \\ -1 & 2 \\ 0 & 3 \end{pmatrix}$$

$$\mathbf{a} \leftarrow \mathbf{matrix}(\mathbf{seq}(-3,1,2),2,3): \begin{pmatrix} -3 & 1 & -1 \\ -1 & -3 & 1 \end{pmatrix}$$

$$\mathbf{a} \leftarrow \mathbf{matrix}(\mathbf{seq}(-3,1,2),2,3,\mathbf{byrow}=\mathbf{TRUE}): \begin{pmatrix} -3 & -1 & 1 \\ -3 & -1 & 1 \end{pmatrix}$$

Para acessar determinado valor de uma matriz, faz-se uso da sintaxe com colchetes, analogamente ao vetor. Por exemplo, para acessar o valor da primeira linha e da segunda coluna da matriz  $\mathbf{a}$ , utiliza-se o comando  $\mathbf{a}[1,2]$ .

Para se obter apenas uma linha ou uma coluna basta usar a mesma sintaxe dos colchetes, porém o segundo ou o primeiro parâmetro, respectivamente, permanece em branco. Assim, a primeira linha da matriz  $\mathbf{a}$  é obtida por  $\mathbf{a}[1, ]$ , bem como a primeira coluna é dada por  $\mathbf{a}[ ,1]$ .

Ainda é possível selecionar somente algumas linhas (ou colunas), o que pode ser chamado de sub-matriz. Com o exemplo abaixo é fácil ver como funciona o processo.

Seja a matriz criada no R por meio do comando  $\mathbf{M} \leftarrow \mathbf{matrix}(\mathbf{seq}(1:10),5,4)$ . Portanto,

$$M = \begin{pmatrix} 1 & 6 & 1 & 6 \\ 2 & 7 & 2 & 7 \\ 3 & 8 & 3 & 8 \\ 4 & 9 & 4 & 9 \\ 5 & 10 & 5 & 10 \end{pmatrix}$$

O resultado do comando  $\mathbf{M}[2:4, 1:3]$ , que seleciona as linhas de 2 a 4 e as colunas de 1 a 3, é:

$$\begin{pmatrix} 2 & 7 & 2 \\ 3 & 8 & 3 \\ 4 & 9 & 4 \end{pmatrix}$$

Uma flexibilidade do R é a possibilidade de fazer esses filtros em linhas e colunas também com negativos. Nesse caso, são retiradas as linhas e colunas selecionadas. Utilizando a mesma matriz  $\mathbf{M}$  do exemplo acima, o resultado do comando  $\mathbf{M}[-1:-3, -2:-3]$  (ou  $\mathbf{M}[-3:-1, -3:-2]$ ), que retira as linhas de 1 a 3 e as colunas de 2 a 3, é:

$$\begin{pmatrix} 4 & 9 \\ 5 & 10 \end{pmatrix}$$

As operações aritméticas com matrizes são similares às operações com vetores. A multiplicação por um escalar é feita em todos os elementos da matriz, conforme exemplo a seguir.

$$(-3) * \begin{pmatrix} -3 & -1 & 1 \\ -3 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 9 & 3 & -3 \\ 9 & 3 & -3 \end{pmatrix}$$

Outras funções importantes de matrizes que, além das tradicionais **sum()**, **mean()**, **median()**, **min()**, **max()**, **range()**, **sd()** e **summary()**, recebem uma matriz como parâmetro de entrada, são:

- **t()**: retorna a matriz transposta
- **diag(n)**: cria a matriz identidade  $n \times n$
- **det()**: calcula o determinante da matriz
- **dim()**: retorna a dimensão da matriz
- **nrow()**: retorna o número de linhas da matriz
- **ncol()**: retorna o número de colunas da matriz
- **rowSums()**: retorna a soma de cada linha
- **rowMeans()**: retorna a média de cada linha
- **colSums()**: retorna a soma de cada coluna
- **colMeans()**: retorna a média de cada coluna
- **sort()**: retorna vetor com os elementos da matriz em ordem crescente
- **is.na()**: para cada posição da matriz, retorna TRUE ou FALSE, sendo o último o caso de a posição correspondente não estar disponível. Isso ocorre quando há um erro e o R atribui o valor **NA**

A soma e a subtração de matrizes são feitas utilizando-se a sintaxe normal, apenas com os sinais “+” e “-” respectivamente, da mesma forma que a som de vetores e números.

Já a multiplicação de matrizes  $A * B$  deve ser feita quando estiver bem definida ( $\#colunas(A) = \#linhas(B)$ ). Além disso, deve-se tomar cuidado com um detalhe da sintaxe do R para a multiplicação de matrizes. Ao contrário dos números reais, em que

basta utilizar o sinal “\*” entre os números, para realizar a multiplicação  $A * B$  (definida da mesma forma que na Matemática), é preciso executar o comando **A %\*\*% B**.

O comando **A \* B** só está definido para matrizes de mesma dimensão e retorna uma outra matriz C de dimensão igual à das matrizes A e B tal que cada elemento  $C(i, j)$  é a multiplicação simples dos elementos  $A(i, j)$  e  $B(i, j)$ . Esta não é a multiplicação usual de matrizes, mas sim o chamado Produto de Hadamard.

## 3 *Inpainting*

Este capítulo trará detalhes da técnica de reconstrução de imagens chamada *inpainting*. Aqui será definida a técnica e serão apresentadas sua história, métodos existentes, vantagens e desvantagens de cada método, aplicações e alternativas.

Ainda neste capítulo é apresentada a biblioteca **imager** em R que realiza a reconstrução de imagens utilizando a técnica de *inpainting*, bem como o programa específico a ser desenvolvido em sala com alunos do Ensino Médio. Esse programa recebe uma foto de entrada, insere algumas faltas de *bits* nela e, posteriormente, a imagem é reconstruída.

### 3.1 O *Inpainting*

O *inpainting* é um processo de recuperação de imagens deterioradas, seja pelo tempo ou com ruídos ou com porções faltando. Também é utilizado na retirada de determinados objetos em imagens, reconstruindo o cenário como se não estivessem nela.

A reconstrução de imagens é uma técnica amplamente usada na restauração de imagens antigas, principalmente de pinturas que foram feitas há séculos, muito presentes em igrejas e quadros, de forma que se consiga restaurar o passado e preservar a história. Essa necessidade não é nova e há muitos séculos que vem sendo utilizada a restauração de pinturas e esculturas.

O que era feito manualmente, como forma de arte subjetiva, foi sendo incrementado e hoje existem técnicas digitais de restauração que, antes aplicada apenas em objetos físicos, hoje também pode ser utilizada em imagens digitais e vídeos. O objetivo é o mesmo: retirar falhas por meio de adição ou remoção de elementos de imagem, preenchendo ou modificando as áreas danificadas de tal forma que um observador não familiarizado com a imagem seja incapaz de detectar as alterações.

A restauração manual – que também abrange diferentes técnicas – é uma arte milenar utilizada até hoje em esculturas e quadros. Já para o retoque de fotografias digitais e vídeos, pode-se fazer uso do *inpainting* digital.

As figuras 3.1, 3.2 e 3.3 ilustram exemplos de retoque manual em pinturas em que cada profissional aplica uma técnica, muitas vezes a depender de como a obra original fora feita.



Fonte: <http://restauro.maurobandeira.com/>

Figura 3.1 – Restauração manual de pintura 1



Fonte: <http://restauro.maurobandeira.com/>

Figura 3.2 – Restauração manual de pintura 2





Fonte: <http://www.restauracaoprates.com.br/>

Figura 3.3 – Restauração manual de pintura 3

Em todos os casos de restauração manual acima, o propósito é o mesmo: fazer com que a pintura final tenha boa qualidade e seja tão similar quanto possível à pintura original. A definição de boa qualidade, neste caso, permite pequenos defeitos, mas que sejam imperceptíveis ao olho humano e um observador não detecte alterações nem consiga perceber que a imagem passou por retoque.

Em resumo, os métodos dos restauradores manuais de fotografias, pinturas e esculturas seguem os seguintes passos:

1. análise da imagem como um todo
2. identificação do estilo de pintura da vizinhança de cada parte deteriorada
3. transmissão da estrutura da vizinhança de cada pedaço deteriorado para o seu interior, inicialmente prolongando os traços que vêm da vizinhança e da borda
4. pintura do interior no mesmo estilo de pintura do restante com a cor correspondente
5. pequenos retoques e adição de detalhes finais

Com o advento da tecnologia, foi criado o retoque digital – ou *inpainting* – que busca reproduzir a restauração artística no âmbito de imagens digitais e vídeos corrompidos, com ajuda da Matemática.

Além de restaurações de imagens danificadas, algoritmos de *inpainting* podem ser aplicados em outros problemas práticos, tais como: restauração de filmes (remoção de arranhões e manchas), remoção de legendas ou vandalizações (pichações), remoção de objetos indesejados, *zoom* digital e super-resolução.

## 3.2 História

O *inpainting* teve sua versão pioneira com BERTALMIO et al., 2000. Ele implementou numericamente os passos 2, 3 e 4 do processo manual descrito no final da seção anterior a partir da transposição de informações de uma imagem em tons de cinza ao longo de linhas de mesmo nível de cinza na imagem.

O passo 1 do processo manual, porém, não foi traduzido no algoritmo proposto por BERTALMIO et al., 2000 tornando o algoritmo de retoque digital mais eficiente devido à busca de informações para a recuperação da parte danificada apenas numa vizinhança desta região.

O passo final do processo manual é relacionado à técnica de Síntese de Textura que vem combinada ao processo de *inpainting* em muitos trabalhos (inclusive em um trabalho de 2011 de Bertalmio), de forma a ampliar o número de imagens que passam por restauração digital sem que um observador consiga determinar as alterações.

A principal diferença entre os retoques manual e digital, é que, no primeiro, o artista que restaura a figura é o principal responsável pela qualidade do resultado final; já no segundo, o usuário apenas define a imagem e quais são as regiões a serem restauradas, além de (possivelmente) outros parâmetros como tamanho de janela, preferências de técnicas, entre outros.

O trabalho de BERTALMIO et al., 2000 é uma das principais referências para todos os demais trabalhos na área. A partir desse trabalho a área de *inpainting* vem sendo aprimorada nos últimos anos e hoje já existem diversos métodos de retoque digital inspirados no primeiro trabalho.



Fonte: BERTALMIO et al., 2000

Figura 3.4 – Recuperação de imagens feitas por Bertalmio

Hoje em dia existem diversos *softwares* e aplicativos *mobile* que utilizam técnicas avançadas de retoque de imagem e vídeo. Um dos mais famosos é o Adobe Photoshop, amplamente utilizado para suavização de fotografias e retoques que, se feitos por profissionais da área, são imperceptíveis ao olho humano.

### 3.3 Tipos de Técnicas de *Inpainting*

#### 3.3.1 *Inpainting* Estrutural

Os primeiros processos de *inpainting* tinham como objetivo recuperar a forma geométrica dos objetos danificados em uma imagem. Essa classe de processos ficou conhecida como *inpainting* estrutural – também chamada de *inpainting* geométrico – e funciona muito bem em casos nos quais a porção danificada tem bordas e forma geométrica bem definidas, além de mudanças suaves de cores e formas na vizinhança das partes a serem reconstruídas. Essa classe de processos de *inpainting* é aplicada a regiões de retoque que precisam apenas de informação estrutural (ou geométrica) para que a imagem seja recuperada.

BERTALMIO et al., 2000 criou uma técnica de *inpainting* estrutural na qual sugeria a reconstrução de áreas danificadas por meio da transferência recursiva da informação

de sua borda e vizinhança. Isso funciona bem com, por exemplo, partes danificadas pequenas e com pouco gradiente de mudança de cor e forma, ou seja, com pouca informação. Também a técnica é eficiente para reconstruir pequenas porções deterioradas em imagens em preto e branco.

Há ainda métodos manuais de computador (que fazem uso da parte subjetiva do artista), nos quais o usuário usa uma ferramenta de clonagem para copiar partes existentes da imagem e restaurar uma parte danificada. A síntese de textura também pode ser usada para incrementar esse processo.

### 3.3.2 *Inpainting* Textural

A técnica de *inpainting* estrutural, apesar de ter representado uma inovação na área de Processamento e Recuperação de Imagens, não atendia às necessidades mais complexas de retoque digital. Tal necessidade motivou a pesquisa na área e a criação de processo de *inpainting* que faziam uso de métodos de síntese de textura.

Não há uma definição formal de textura de uma imagem e os estudiosos a definem de forma mais qualitativa do que quantitativa. Uma das primeiras definições foi dada por R.M. Haralick em 1979:

“Um imagem com textura pode ser descrita pelo número e pelos tipos de suas primitivas, bem como pela organização espacial ou pelo *layout* de suas primitivas. A organização espacial pode ser aleatória ou pode obedecer um padrão relativamente uniforme entre uma primitiva e suas primitivas vizinhas. Esta dependência pode ser estrutural, probabilística ou funcional.”

L.V. Gool et al. definiram, em 1983, como:

“Textura pode ser definida como uma estrutura composta de uma área contendo um número grande de elementos relativamente ordenados ou padrões que individualmente não representam muito, mas, se observados globalmente, tais elementos podem ser usados para identificar uma determinada área.”

Já B.Julesz., em 1984, definiu da seguinte forma:

“Textura é definida como o arranjo espacial e a natureza de componentes estruturais que podem ser pontos individuais em uma imagem ou *textons* mais complexos.”

Resumindo, pode-se dizer que textura é uma propriedade de um *pixel* que o diferencia dos seus *pixels* vizinhos. Portanto, pode-se falar em texturas com diferentes graus

de aspereza, granulidade, suavidade, etc. A restauração digital da textura é um desafio muito mais difícil do que a restauração da geometria dos objetos presentes nas regiões danificadas. Esta classe de técnicas ficou conhecida como *inpainting* textural.

A síntese de texturas pode ser feita de várias formas. As mais utilizadas e mais difundidas são aquelas que contam com o auxílio de modelos: de Markov discretos randômicos, de Markov gaussianos discretos randômicos e de fractais.

Pela falta de consenso na definição e de abordagem formal quanto à textura de uma imagem, o desempenho e a qualidade de muitos algoritmos de recuperação de textura são avaliados com relação ao desempenho humano na realização da mesma tarefa.

### 3.3.3 *Inpainting* Estrutural e Textural

A combinação das duas técnicas anteriores deu origem à técnica de retoque digital mais completa que existe. Ela tenta preencher tanto estrutura (ou seja, forma) como textura faltantes da imagem danificada. É a opção que funciona melhor com imagens mais complexas, de várias cores, formatos e padrões; é por esse motivo que os métodos mais modernos e de ponta combinam técnicas de reparação estrutural e textural.

Basicamente, os métodos combinados se dividem entre pintura baseada em modelo e baseada em exemplos.

A pintura baseada em modelo segue a abordagem Bayesiana. Nela, as informações ausentes são estimadas a partir da combinação dos modelos de sua vizinhanças, bem como dos dados da porção a ser retocada realmente observados. Na linguagem determinística, isso levou a vários modelos variados de pintura.

A pintura de imagem com base em exemplos tenta automatizar o processo da ferramenta de clonagem. Ela preenche falhas na imagem, procurando padrões semelhantes em uma região de origem próxima da imagem e copiando os *pixels* do padrão mais semelhante para a falha. Ao executar o preenchimento no nível do padrão, ao invés de executá-lo no nível de *pixels*, o algoritmo reduz os erros de desfoque causados por técnicas que substituem *pixel* a *pixel*.

Neste âmbito, foram desenvolvidos métodos utilizando equações diferenciais com condições de contorno de Dirichlet para garantir a continuidade, fazendo, assim, com que a imagem retocada seja, aparentemente, uniforme e suave. Isso funciona bem se as informações ausentes estiverem na parte homogênea de uma área de objeto.

Há outros métodos que analisam a frequência da imagem para reproduzir a textura. HIRANI; TOTSUKA, 1996 desenvolveram um método simples e eficaz para preencher regiões com uma textura selecionada combinando informações do domínio espacial com o domínio da frequência. Neste caso, o usuário seleciona a região que tem textura similar

à da região danificada para que o sistema replique. Caso a região a ser preenchida tenha várias texturas distintas, é preciso segmentá-la e encontrar texturas similares na imagem para cada porção.

O algoritmo desenvolvido por BERTALMIO et al., 2003 decompõe a imagem como a soma de duas imagens diferentes: a primeira carrega as informações estruturais da imagem original e a segunda determina a sua textura. Então, é feita a reconstrução separadamente pelos métodos de retoque estrutural e textural, respectivamente, aplicando-se, ao final, a soma das duas imagens reconstruídas para gerar o retoque da imagem original.

Diferentemente do algoritmo anterior, CRIMINISI; PEREZ; TOYAMA, 2004 propuseram um algoritmo que trata simultaneamente informações estrutural e textural. Fixado um *pixel* da borda da região de retoque, o algoritmo busca uma janela da imagem de tamanho escolhido pelo usuário tal que a janela não possua pontos da região danificada e seja a mais próxima da janela de mesmo tamanho centrada no *pixel* fixado. A partir daí, o valor de cada *pixel* desta última é substituído pelo valor do *pixel* correspondente na janela mais próxima da que está com pontos a serem reconstruídos. A vantagem desse método é que, de uma só vez, ele faz o retoque digital de forma e textura.

Investigações recentes incluíram a exploração das propriedades da transformada *wavelet* para realizar pintura no domínio da frequência espacial; tais técnicas têm obtido um melhor desempenho quando comparadas às técnicas de pintura com base na frequência.

### 3.4 Ruídos

Um ruído de uma fotografia é o equivalente digital dos grãos dos filmes utilizados em câmeras analógicas. Outra analogia que se pode fazer é com o sutil chiado que se percebe quando uma música é ouvida em um volume muito alto. Em imagens digitais, esse ruído aparece como manchas ou pontos aleatórios em uma superfície originalmente suave e pode degradar a qualidade da imagem.

Os ruídos podem ser causados por falta de brilho ou por falsas informações embutidas na imagem ou no vídeo, muitas vezes inseridas ali no processo de transmissão entre a realidade e a fotografia ou o vídeo.

O ruído é proporcional à sensibilidade escolhida, à duração da exposição e à temperatura. Ainda no caso das fotografias, a quantidade de ruído também depende da câmera utilizada para gerar a imagem, já que câmeras profissionais de alto nível costumam ter menos ruído que câmeras compactas comuns ou câmeras de aparelho celular.

Tecnicamente, o ruído é definido como um *pixel* de alta oscilação numa imagem. Esse *pixel* tem características indesejáveis. Presente em muitas fotos, o ruído tem características de cor e brilho diferentes da realidade e, portanto, seria interessante conseguir

retirá-lo para uma foto mais fidedigna.

De acordo com SEARA, 1998 toda imagem adquirida no ambiente digital é propícia a algum tipo de ruído. Ruídos são, portanto, inevitáveis e imprevisíveis nas imagens digitais. Nas imagens existem dois tipos de ruídos:

- Ruído Gaussiano
- Ruído “Sal-e-Pimenta”

O primeiro é um ruído que apresenta uma distribuição de Gauss. É um ruído estatístico que tem uma função de densidade de probabilidade e surge devido à má iluminação, por conta de altas temperaturas ou problemas na transmissão da imagem, deixando-a até mesmo extremamente danificada, conforme constataram KLEIN; GALLAGER, 2001.

O ruído “Sal-e-Pimenta” ocorre a partir de erros na transmissão de dados. Com a transmissão da imagem, algum *pixel* pode se corromper, alterando os tons de cinza de cada *pixel* vizinho, conforme RIBEIRO, 2006.

Vale lembrar que há casos em que o ruído pode ser desejado, no caso de filtros que têm o propósito de envelhecer imagens ou inserir artefatos. Nesta seção, no entanto, como o objetivo é a remoção do ruído, focar-se-á na sua definição de ser indesejado à medida que polui a imagem.



Fonte: CONTRERAS; BOAVENTURA, 1998 e FILHO; NETO, 1999

Figura 3.5 – Imagem com ruído “Sal-e-Pimenta” antes e depois de sua remoção por filtro da mediana

A Figura 3.5 (CONTRERAS; BOAVENTURA, 1998 e FILHO; NETO, 1999) representa um dos ruídos mais populares em imagens digitais, o chamado ruído “Sal-e-Pimenta”, comum na transmissão de dados. Sua remoção pode ser feita usando-se um filtro chamado filtro da mediana. Na realidade, como citado no início deste capítulo no contexto do *inpainting*, o objetivo é atenuar o efeito do ruído, de forma que um observador humano não perceba que a imagem da direita – após o filtro – tenha passado por algum método de reconstrução de imagem.

## 3.5 Recuperação de Imagens

O algoritmo de recuperação de imagens do R possui um funcionamento simples. Ele é baseado na filtragem da imagem danificada com um filtro Gaussiano. Esta seção tem como objetivo descrever o funcionamento deste tipo de abordagem e a sua qualidade.

### 3.5.1 Equação do Calor e Recuperação de Imagens

Apesar de, em um primeiro momento, parecer pouco intuitivo, existe uma ligação próxima entre as equações diferenciais parciais e a recuperação de imagens. No trabalho pioneiro em *inpainting*, BERTALMIO et al., 2000 utilizou equações diferenciais parciais (EDP) de terceira ordem para recuperar imagens. O algoritmo consiste em transportar progressivamente as linhas de nível (ou isófotos) que incidem na fronteira da região a ser reconstruída de forma que o ângulo de interseção seja preservado, impedindo o cruzamento das linhas de nível. Para isto, o algoritmo aplica um filtro de difusão anisotrópica para corrigir a direção do retoque. Desta forma, a região danificada recebe informações da sua vizinhança e é reconstruída de forma suavizada.

Vale definir aqui o que são isófotos: em uma imagem com escalas de cinza, representam fronteiras definidas por *pixels* com o mesmo tom de cinza.

O método utilizado neste trabalho para retoque digital é mais simples, baseado na equação do calor, vastamente estudada e utilizada em modelagens na Matemática e na Física. A equação do calor surge no contexto de modelagem da difusão do calor em sólidos e aparece na literatura nas mais diversas versões. Segundo EVANS, 2010, a equação a seguir, e que será utilizada nesse trabalho, é a sua versão bidimensional:

$$\frac{\partial u}{\partial t} = \Delta u, \quad (5.1)$$

em que  $\mathbf{x} = (x, y)$ ,  $u(\mathbf{x}, t)$  é a temperatura na posição espacial  $\mathbf{x}$  do objeto, no tempo  $t$  e  $\Delta u$  é o operador Laplaciano

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}. \quad (5.2)$$

Observa-se que será utilizada a versão bidimensional da equação do calor, pois imagens digitais podem ser entendidas como funções  $h : U \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ , em que  $h$  é representa o nível de luminosidade em cada ponto  $\mathbf{x} \in U$ .

Associadas à equação (5.1), são fornecidas as condições iniciais e de contorno (ou fronteira) do problema que são usadas para a obtenção da solução geral. Aqui,  $t \geq 0$  e  $\mathbf{x} \in U$ ,  $U$  aberto, onde  $U \subset \mathbb{R}^2$ . Portanto, procura-se encontrar a solução  $u : U \times [0, \infty) \rightarrow \mathbb{R}$ ,  $u = u(\mathbf{x}, t)$ , sujeita às condições iniciais e de contorno

$$u(\mathbf{x}, 0) = f(\mathbf{x}), \quad \mathbf{x} \in U \quad (5.3)$$

$$u(\mathbf{x}, t) = g(\mathbf{x}, t), \quad \mathbf{x} \in \partial U. \quad (5.4)$$



Segundo EVANS, 2010, mais genericamente, a equação do calor – também chamada de equação da difusão – descreve a evolução, ao longo do tempo, da densidade  $u$  de alguma grandeza como calor ou concentração química. Se  $V \subset U$  é uma região suave, então a taxa de variação da grandeza na região  $V$  é igual a menos o fluxo líquido ao longo da fronteira  $\partial V$ .

$$\frac{d}{dt} \int_V u \, d\mathbf{x} = - \int_{\partial V} \mathbf{F} \cdot \nu \, dS,$$

onde  $\mathbf{F} = \mathbf{F}_x \mathbf{i} + \mathbf{F}_y \mathbf{j}$  é o campo vetorial de densidade do fluxo e  $\nu$  é o vetor normal exterior a  $\partial V$ . Como a região  $V$  é arbitrária, conclui-se que

$$\frac{\partial u}{\partial t} = - \operatorname{div} \mathbf{F}, \quad (5.5)$$

onde  $\operatorname{div} \mathbf{F}$  representa o divergente do campo vetorial  $\mathbf{F}$ , isto é,

$$\operatorname{div} \mathbf{F} = \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y}.$$

Em diversas situações o campo vetorial  $\mathbf{F}$  é diretamente proporcional (porém no sentido oposto) ao gradiente de  $u$ , isto é,  $\mathbf{F} = -a \nabla u$ , com  $a$  real positivo. Vale lembrar que

$$\nabla u = \frac{\partial u}{\partial x} \mathbf{i} + \frac{\partial u}{\partial y} \mathbf{j}.$$

Substituindo este resultado na equação (5.5), gera-se a EDP a seguir:

$$\frac{\partial u}{\partial t} = a \operatorname{div}(\nabla u) = a \Delta u \quad (5.6)$$

Uma das formas de encontrar soluções para uma EDP é através da busca por soluções fundamentais, isto é, soluções que serão base para soluções de distintos problemas que podem ser estudados. Aqui será buscada a solução fundamental para o problema do calor dado pela equação diferencial (5.1) e sujeita apenas à condição inicial (5.3), ambas definidas em todo  $\mathbb{R}^2$ .

Neste sentido, é imediato observar que se  $u(\mathbf{x}, t)$  é solução da equação (5.1), então  $u(\lambda \mathbf{x}, \lambda^2 t)$  também é solução,  $\forall \lambda \in \mathbb{R}$ . Dentre todas essas soluções, é constante a razão  $\frac{|\mathbf{x}|^2}{t}$ , o que leva à busca de uma solução da forma  $u(\mathbf{x}, t) = v\left(\frac{|\mathbf{x}|^2}{t}\right)$ , para  $t > 0$  e  $\mathbf{x} \in \mathbb{R}^2$ , com a função  $v$  a ser determinada.

Para encontrar a chamada solução fundamental da equação do calor, EVANS, 2010 sugere buscar soluções mais gerais na forma:

$$u(\mathbf{x}, t) = \frac{1}{t^\alpha} v\left(\frac{\mathbf{x}}{t^\beta}\right) \quad (t > 0, \mathbf{x} \in \mathbb{R}^2), \quad (5.7)$$

onde as incógnitas são  $\alpha$ ,  $\beta$  e a função  $v : \mathbb{R}^2 \rightarrow \mathbb{R}$ . Tais soluções surgem quando se buscam soluções invariantes à propriedade da dilatação, isto é,

$$u(\mathbf{x}, t) = \lambda^\alpha u(\lambda^\beta \mathbf{x}, \lambda t) \quad \forall \lambda > 0, t > 0, \mathbf{x} \in \mathbb{R}^2,$$

com  $\lambda = \frac{1}{t}$  e  $v(\mathbf{y}) = u(\mathbf{y}, 1)$ .

Substituindo (5.7), que equivale a  $u(\mathbf{x}, t) = t^{-\alpha} u(t^{-\beta} \mathbf{x}, 1)$ , na equação do calor homogênea (5.1):

$$-\alpha t^{-(\alpha+1)} v(\mathbf{y}) + \beta t^{-(\alpha+1)} \mathbf{y} \cdot \nabla v(\mathbf{y}) + t^{-(\alpha+2\beta)} \Delta v(\mathbf{y}) = 0, \quad (5.8)$$

onde  $\mathbf{y} = t^{-\beta} \mathbf{x}$ .

Para simplificar o fator  $t^{-(\alpha+1)}$  e ter uma expressão apenas em função de  $v = v(\mathbf{y})$ , faz-se  $\beta = \frac{1}{2}$ , obtendo

$$-\alpha v + \frac{1}{2} \mathbf{y} \cdot \nabla v + \Delta v = 0 \quad (5.9)$$

Com o intuito de simplificar ainda mais a expressão encontrada na equação (5.9), supõe-se que  $v$  é uma função radial, isto é,  $v(\mathbf{y}) = w(|\mathbf{y}|)$  para alguma função  $w : \mathbb{R} \rightarrow \mathbb{R}$ . Desta forma, a equação (5.9) pode ser reescrita como

$$\alpha w + \frac{1}{2} r w' + w'' + \frac{1}{r} w' = 0, \quad (5.10)$$

em que  $r = |\mathbf{y}|$  e as derivadas agora são em função da variável  $r$ .

A solução da equação (5.10) com  $\alpha = 1$  pode ser obtida considerando as condições  $\lim_{r \rightarrow \infty} w = \lim_{r \rightarrow \infty} w' = 0$ . Vale observar que a constante  $\alpha$  depende da dimensão do espaço considerado, a saber,  $\alpha = \frac{n}{2}$ ; neste caso, como se trata do  $\mathbb{R}^2$ , a constante  $n = 2$  e portanto  $\alpha = 1$ . Finalmente, a solução da equação (5.10) fornece o que é conhecido como solução fundamental da equação do calor:

$$\Phi(\mathbf{x}, t) = \frac{1}{4\pi t} e^{-\frac{|\mathbf{x}|^2}{4t}}, \quad \text{se } \mathbf{x} \in \mathbb{R}^2, t > 0. \quad (5.11)$$

A escolha da constante  $(4\pi)^{-1}$  foi feita para que a propriedade abaixo seja satisfeita

$$\int_{\mathbb{R}^2} \Phi(\mathbf{x}, t) d\mathbf{x} = 1, \quad (5.12)$$

para cada  $t > 0$ .

Vale ressaltar que, apesar de  $\mathbf{x} \in \mathbb{R}^2$ , a solução fundamental – também chamada núcleo Gaussiano – é radial, ou seja, utiliza para o cálculo da imagem da função o módulo do vetor  $\mathbf{x} = (x, y)$ .

Na origem do sistema, i.e., no ponto  $(\mathbf{0}, 0)$ , a solução fundamental é singular. Logo, considerando a equação (5.1) definida em  $\mathbb{R}^2 \times (0, \infty)$ , fora da origem é imediato ver que a equação (5.11) é solução da equação do calor (5.1). Além disso, para cada  $\mathbf{y} \in \mathbb{R}^2$  fixado é possível observar que  $\Phi(\mathbf{x} - \mathbf{y}, t)$  também é solução da equação (5.1) em  $\mathbb{R}^2 \times (0, \infty)$ .

Finalmente, considerando uma condição inicial  $u(\mathbf{x}, 0) = f(\mathbf{x})$ , definida em todo o  $\mathbb{R}^2$ , pode-se mostrar que a solução  $u(\mathbf{x}, t)$  do problema de valor inicial é, de fato, a convolução da solução fundamental com a condição inicial  $u(\mathbf{x}, 0) = f(\mathbf{x})$ ; isto é,

$$\begin{aligned} u(\mathbf{x}, t) &= \int_{\mathbb{R}^2} \Phi(\mathbf{x} - \mathbf{y}, t) f(\mathbf{y}) d\mathbf{y} \\ &= \frac{1}{4\pi t} \int_{\mathbb{R}^2} e^{-\frac{|\mathbf{x} - \mathbf{y}|^2}{4t}} f(\mathbf{y}) d\mathbf{y} \end{aligned} \quad (5.13)$$

é a solução do problema de valor inicial descrito (EVANS, 2010).

A solução dada pela equação (5.13) é de fundamental importância neste trabalho, pois ela é a base do algoritmo de recuperação de imagens implementado em R. Neste algoritmo, as regiões danificadas da imagem são recuperadas através da convolução da imagem por um núcleo Gaussiano, como o da equação (5.13). Assim, as regiões defeituosas são substituídas por novos valores que são a média dos valores próximos a eles. Esse procedimento será melhor descrito na Subseção 3.6.1.

Finalmente, observa-se que é também possível mostrar que a solução do problema de valor inicial associado à equação do calor é dada pela equação (5.13) utilizando a transformada de Fourier. É possível encontrar essa abordagem nas referências FIGUEIREDO, 2018; JUNIOR; IORIO, 2013; EVANS, 2010.

### 3.5.2 Precisão do Método

Para estimar a precisão do método de *inpainting* descrito na Subseção 3.5.1 será estudada uma variação do problema do calor. Quando o tempo tende a infinito, ou seja, no estado estacionário, a equação do calor transforma-se na equação de Laplace, visto que a temperatura não varia mais, ou seja,  $\frac{\partial u}{\partial t} = 0$  quando  $t \rightarrow \infty$ . Neste caso, obtém-se

$$\Delta u \Big|_{t \rightarrow \infty} = \frac{\partial u}{\partial t} \Big|_{t \rightarrow \infty} \Rightarrow \Delta u = 0$$

O método de recuperação de imagens que utiliza a equação de Laplace é o *inpainting* harmônico que, segundo SCHONLIEB, 2015, é um dos métodos mais simples e fáceis de se entender para interpolação de imagem. Nesse método, a imagem reconstruída é calculada como a solução da equação de Laplace, que vem a ser um processo de interpolação linear suave que substitui as falhas por uma média dos valores vizinhos.

Vale lembrar a definição de espaço  $L^2(\Omega)$ , que é um espaço de Hilbert (ou seja, um espaço vetorial com a norma canônica definida por  $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ , ou seja, pela raiz quadrada do produto interno de um vetor por ele mesmo) dotado do seguinte produto interno:

$$\langle f, g \rangle = \int_{\Omega} \overline{f(x)} g(x) dx.$$

A formulação do *inpainting* com equação de Laplace é feita da seguinte forma em SCHONLIEB, 2015:

**Definição 3.5.1.** *Sejam  $g \in L^2(\Omega)$  a imagem inicial a ser aplicado o *inpainting* e  $D \subset \Omega$  a região danificada. Supondo a fronteira de  $D$  como contínua e como uma curva simples de Jordan, a imagem  $u \in H^1(D)$  é a solução fraca – ou seja, descontínua – da equação de Laplace*

$$\begin{aligned}\Delta u &= 0 \text{ em } D \\ u &= g \text{ em } \partial D\end{aligned}\tag{5.14}$$

Ou seja,  $u$  satisfaz

$$\begin{aligned}\int_D \nabla u \cdot \nabla \psi \, d\mathbf{x} &= 0, \quad \forall \psi \in H_0^1(D), \\ u &= g \text{ em } \partial D\end{aligned}\tag{5.15}$$

A solução fraca de (5.14) é chamada por SCHONLIEB, 2015 de **extensão harmônica**  $u$  de  $g$  em  $\Omega - D$  para  $D$ . A qualidade de extensões harmônicas para funções suaves  $g \in C^2(\Omega)$  foi estudada por CHAN; SHEN, 2002 com a aplicação da segunda fórmula de Green.

No contexto deste trabalho, bem como em CHAN; SHEN, 2002, o desenvolvimento será feito para o espaço de duas dimensões. Para estudar a precisão do método por meio da fórmula de Green, é necessário que a imagem desejada seja suave ou que o domínio a ser aplicada a técnica de *inpainting* esteja contido em um espaço 2-D suave. A partir deste caso mais comportado é possível avançar para o desenvolvimento de técnicas que sirvam para casos mais complexos com singularidades.

Seja  $u^0$  a imagem suave a ser recuperada, definida em um domínio bidimensional  $\Omega$  e seja  $D$ , com diâmetro  $d$ , a região com as falhas. O que se busca com a reconstrução da imagem é uma boa aproximação  $u_D$  de  $u^0|_D$  (isto é,  $u^0$  restrita a  $D$ ).

No caso de funções de uma variável, as funções harmônicas são as lineares. Para o caso bidimensional, CHAN; SHEN, 2002 propõem usar a segunda fórmula de Green em  $D$ :

$$\int_D (u\Delta v - v\Delta u) \, d\mathbf{x} = \int_{\partial D} \left( u \frac{\partial v}{\partial \mathbf{n}} - v \frac{\partial u}{\partial \mathbf{n}} \right) ds,\tag{5.16}$$

onde  $u$  e  $v$  são funções  $C^2$  definidas no fecho de  $D$ ,  $\mathbf{n}$  é o vetor normal a  $\partial D$  (sentido para fora) e  $s$  representa o comprimento de arco.

*Demonstração da Segunda fórmula de Green.* É possível demonstrar (5.16) por meio da aplicação do Teorema do Divergente à função  $u\nabla v$ . Primeiramente, tem-se que

$$\operatorname{div}(u\nabla v) = u\Delta v + \nabla u \cdot \nabla v$$

Logo,

$$\int_D \operatorname{div}(u\nabla v) \, d\mathbf{x} = \int_D u\Delta v \, d\mathbf{x} + \int_D \nabla u \cdot \nabla v \, d\mathbf{x} = \int_{\partial D} u\nabla v \cdot \mathbf{n} \, ds,$$

onde a igualdade entre a primeira e a última integrais constitui o Teorema do Divergente. Rearranjando, deduz-se a primeira fórmula de Green em  $D$ .

$$\int_D \nabla u \cdot \nabla v \, d\mathbf{x} = - \int_D u\Delta v \, d\mathbf{x} + \int_{\partial D} u\nabla v \cdot \mathbf{n} \, ds$$

Aplicando o Teorema do Divergente à função  $(u\nabla v - v\nabla u)$ :

$$\int_D \operatorname{div}(u\nabla v - v\nabla u) \, d\mathbf{x} = \int_D (u\Delta v - v\Delta u) \, d\mathbf{x} = \int_{\partial D} (u\nabla v - v\nabla u) \cdot \mathbf{n} \, ds$$

Portanto,

$$\int_D (u\Delta v - v\Delta u) \, d\mathbf{x} = \int_{\partial D} \left( u \frac{\partial v}{\partial \mathbf{n}} - v \frac{\partial u}{\partial \mathbf{n}} \right) \, ds$$

□

Seja  $G(\mathbf{x}_0, \mathbf{x})$  a função de Green para a equação de Poisson em  $D$  com condição de fronteira nula em  $\partial D$ . Ou seja, para qualquer ponto fixado  $\mathbf{x}_0 = (x_0, y_0) \in D$  – que seria o equivalente à “fonte” – com  $\mathbf{x} = (x, y) \in D$  sendo o equivalente ao “campo vetorial” nos termos da equação de Poisson, tem-se que  $G(\mathbf{x}_0, \mathbf{x})$  satisfaz

$$-\Delta G = \delta(\mathbf{x} - \mathbf{x}_0) \text{ e } G \Big|_{\partial D} = 0, \tag{5.17}$$

onde  $\delta(\cdot)$  é a função Delta de Dirac que possui a seguinte propriedade (chamada propriedade da “peneira”):

$$\int_{\mathbb{R}^2} \delta(\mathbf{x} - \mathbf{x}_0) f(\mathbf{x}) \, d\mathbf{x} = f(\mathbf{x}_0).$$

Aplicando a segunda fórmula de Green (5.16) para  $u = u^0(\mathbf{x})$  e  $v = -G(\mathbf{x}_0, \mathbf{x})$ :

$$\begin{aligned} \int_D (u^0(\mathbf{x})\Delta(-G(\mathbf{x}_0, \mathbf{x})) - (-G(\mathbf{x}_0, \mathbf{x}))\Delta(u^0(\mathbf{x}))) \, d\mathbf{x} = \\ \int_{\partial D} \left( u^0(\mathbf{x}) \frac{\partial(-G(\mathbf{x}_0, \mathbf{x}))}{\partial \mathbf{n}} - (-G(\mathbf{x}_0, \mathbf{x})) \frac{\partial u^0(\mathbf{x})}{\partial \mathbf{n}} \right) \, ds, \end{aligned}$$

ou seja,

$$\begin{aligned} \int_D u^0(\mathbf{x}) \overbrace{\Delta(-G(\mathbf{x}_0, \mathbf{x}))}^{\delta(\mathbf{x} - \mathbf{x}_0)} \, d\mathbf{x} - \int_{\partial D} \overbrace{G(\mathbf{x}_0, \mathbf{x})}^0 \frac{\partial u^0(\mathbf{x})}{\partial \mathbf{n}} \, ds = \\ \int_{\partial D} u^0(\mathbf{x}) \frac{\partial(-G(\mathbf{x}_0, \mathbf{x}))}{\partial \mathbf{n}} \, ds + \int_D G(\mathbf{x}_0, \mathbf{x}) \Delta(-u^0(\mathbf{x})) \, d\mathbf{x}. \end{aligned}$$

Assim

$$\int_D u^0(\mathbf{x})\delta(\mathbf{x} - \mathbf{x}_0)dz = \int_{\partial D} u^0(\mathbf{x})\frac{\partial(-G(\mathbf{x}_0, \mathbf{x}))}{\partial \mathbf{n}}ds + \int_D G(\mathbf{x}_0, \mathbf{x})\Delta(-u^0(\mathbf{x}))d\mathbf{x}.$$

Finalmente,

$$u^0(\mathbf{x}_0) = \int_{\partial D} u^0(\mathbf{x})\frac{\partial(-G(\mathbf{x}_0, \mathbf{x}))}{\partial \mathbf{n}}ds + \int_D G(\mathbf{x}_0, \mathbf{x})\Delta(-u^0(\mathbf{x}))d\mathbf{x}. \quad (5.18)$$

Em (5.18), denomine o primeiro termo do lado direito de  $u^h(\mathbf{x}_0)$  e o segundo termo do lado direito de  $u^a(\mathbf{x}_0)$ , denotando, respectivamente, as extensões harmônica e anti-harmônica de  $g = u^0|_{\partial D}$ .

A parcela anti-harmônica  $u^a(\mathbf{x}_0)$  é solução da equação de Poisson

$$\Delta u^a(\mathbf{x}) = \Delta u^0(\mathbf{x}), \quad \mathbf{x} \in D \quad \text{e} \quad u^a|_{\partial D} = 0$$

Para estabelecer a precisão do método de recuperação em imagens suaves, é utilizada a função de Green aplicada ao domínio bidimensional. Os resultados a seguir contribuem para o estudo da precisão de métodos que fazem uso de EDP, em particular, por meio da equação de Laplace.

**Teorema 3.5.2.** *Sejam  $d = \max_{x,y \in \partial D} |x - y|$  o diâmetro de um domínio  $D$  e  $G(\mathbf{x}_0, \mathbf{x})$  a função de Green associada à equação de Poisson (5.17). Então*

$$\int_D G(\mathbf{x}_0, \mathbf{x}) d\mathbf{x} \leq \frac{d^2}{4}.$$

A demonstração do Teorema acima é feita em CHAN; SHEN, 2002 com base nos dois lemas a seguir.

**Lema 3.5.3.** *(Lema da comparação) Sejam  $D_1 \subset D_2$  com suas funções de Green correspondentes  $G_1(\mathbf{x}_0, \mathbf{x})$  e  $G_2(\mathbf{x}_0, \mathbf{x})$ . Então,  $\forall \mathbf{x}_0, \mathbf{x} \in D_1$ ,*

$$G_1(\mathbf{x}_0, \mathbf{x}) \leq G_2(\mathbf{x}_0, \mathbf{x}).$$

*Demonstração.* Para iniciar esta demonstração, vale lembrar que toda função de Green com condição de fronteira nula é não negativa. Para  $\mathbf{x}_0 \in D_1$ , defina

$$g(\mathbf{x}) = G_2(\mathbf{x}_0, \mathbf{x}) - G_1(\mathbf{x}_0, \mathbf{x}).$$

Então, ao longo da fronteira de  $D_1$ , como  $G_1(\mathbf{x}_0, \mathbf{x}) = 0$ ,

$$g(\mathbf{x}) = G_2(\mathbf{x}_0, \mathbf{x}) \geq 0.$$

Além disso, segundo CHAN; SHEN, 2002, a função  $g(\mathbf{x})$  é harmônica no interior de  $D_1$ . Pelo princípio do máximo-mínimo em funções harmônicas (GILBARG; TRUDINGER, 2001),  $g(\mathbf{x})$  assume seu valor mínimo na fronteira, levando à conclusão que  $g(\mathbf{x}) \geq 0$  para todo  $\mathbf{x} \in D_1$  e, portanto, provando o lema.  $\square$

**Lema 3.5.4.** *Seja  $B_1$  o disco unitário centrado na origem e  $G_1(\mathbf{x}_0, \mathbf{x})$  sua função de Green. Então,  $\forall \mathbf{x}_0 \in B_1$ ,*

$$\int_{B_1} G_1(\mathbf{x}_0, \mathbf{x}) d\mathbf{x} = \frac{1 - |\mathbf{x}_0|^2}{4}$$

*Demonstração.* Considere a equação de Poisson em  $B_1$

$$-\Delta u = 1, \quad u \Big|_{\partial B_1} = 0. \quad (5.19)$$

Considerando o caráter circular do domínio, é conveniente resolver o problema dado pela equação (5.19) em coordenadas polares. Neste caso, o operador Laplaciano em coordenadas polares é dado por

$$\Delta u = \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right),$$

em que  $r = |\mathbf{x}|$ .

Aqui procura-se por uma solução radial (isto é, dependente somente de  $r$ ). Portanto, pode-se supor que  $u = u(r)$ , logo,  $\frac{\partial^2 u}{\partial \theta^2} = 0$ . Como  $\Delta u = -1$ , então

$$-1 = \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right),$$

isto é,

$$-r = \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right).$$

Integrando ambos os lados,

$$\int -r dr = \int \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right) dr = r \frac{\partial u}{\partial r}.$$

Logo,

$$-\frac{r^2}{2} + C = r \frac{\partial u}{\partial r}$$

e, portanto,

$$-\frac{r}{2} + \frac{C}{r} = \frac{\partial u}{\partial r}.$$

Finalmente, integrando mais uma vez com relação à variável  $r$ :

$$u(r) = -\frac{r^2}{4} + C \ln r + D,$$

para constantes reais  $C$  e  $D$ .

Como  $u(r)$  é uma função que deve ser finita na origem, conclui-se que  $C = 0$ . Para encontrar o valor da constante  $D$ , basta utilizar a condição de contorno  $u(r = 1) = 0$ , visto que a função  $u$  vale zero na fronteira de  $B_1$ .

$$0 = u(1) = -\frac{1}{4} + D \Leftrightarrow D = \frac{1}{4}$$

Conclui-se, portanto, que a solução de (5.19) é

$$u(\mathbf{x}) = \frac{1 - |\mathbf{x}|^2}{4} = \frac{1 - x^2 - y^2}{4}.$$

Por outro lado, por (5.18),

$$u(\mathbf{x}_0) = \int_{\partial B_1} \cancel{u^0(\mathbf{x})} \frac{\partial(-G_1(\mathbf{x}_0, \mathbf{x}))}{\partial \mathbf{n}} ds + \int_{B_1} G_1(\mathbf{x}_0, \mathbf{x}) \cancel{(-\Delta u(\mathbf{x}))} d\mathbf{x} = \int_{B_1} G_1(\mathbf{x}_0, \mathbf{x}) d\mathbf{x}$$

o que prova o resultado.  $\square$

Agora é possível demonstrar o Teorema (3.5.2).

*Demonstração.* Para um ponto fixado  $\mathbf{w} \in D$ , seja  $B_d$  o disco centrado em  $\mathbf{w}$  e raio  $d$ . Então  $D \subset B_d$ . Seja  $G_d(\mathbf{x}_0, \mathbf{x})$  a função de Green para  $B_d$ . Pelo Lema (3.5.3),

$$G(\mathbf{x}_0, \mathbf{x}) \leq G_d(\mathbf{x}_0, \mathbf{x}), \quad \forall \mathbf{x}, \mathbf{x}_0 \in D$$

Por simplicidade, faz-se  $\mathbf{w} = \mathbf{0}$ . Então, pela propriedade de dilatação (válida apenas para o caso bidimensional),

$$G_d(\mathbf{x}_0, \mathbf{x}) = G_1\left(\frac{\mathbf{x}_0}{d}, \frac{\mathbf{x}}{d}\right),$$

onde  $G_1$  é a função de Green para  $B_1$  conforme Lema (3.5.4). Usando o Lema (3.5.4), conclui-se que, para qualquer  $\mathbf{x}_0 \in D$ ,

$$\int_D G(\mathbf{x}_0, \mathbf{x}) d\mathbf{x} \leq \int_D G_d(\mathbf{x}_0, \mathbf{x}) d\mathbf{x} \leq \int_{B_d} G_d(\mathbf{x}_0, \mathbf{x}) d\mathbf{x} = \int_{B_d} G_1\left(\frac{\mathbf{x}_0}{d}, \frac{\mathbf{x}}{d}\right) d\mathbf{x}$$

Fazendo  $\mathbf{x}' = \frac{\mathbf{x}}{d}$  e lembrando que  $d\mathbf{x} = dx dy$ , vem que  $d\mathbf{x}' = dx' dy' = \frac{dx dy}{d^2}$  ou, equivalentemente,  $d\mathbf{x} = d^2 d\mathbf{x}'$  e o lado direito do desenvolvimento acima fica:

$$d^2 \int_{B_1} G_1\left(\frac{\mathbf{x}_0}{d}, \mathbf{x}'\right) d\mathbf{x}' = d^2 \frac{1 - |\mathbf{x}_0/d|^2}{4} \leq \frac{d^2}{4}.$$



O Lema (3.5.4) foi utilizado aqui porque foram feitas as hipóteses de  $\mathbf{w} = \mathbf{0} \in D$  e  $\mathbf{x}_0 \in D$ . Se não for o caso, então basta substituir  $\mathbf{x}_0$  por  $(\mathbf{x}_0 - \mathbf{w})$  e  $\mathbf{x}$  por  $(\mathbf{x} - \mathbf{w})$ , completando a demonstração.  $\square$

Baseado no Teorema (3.5.2), faz-se a estimativa da precisão de métodos lineares de *inpainting* via extensão harmônica com o auxílio da segunda fórmula de Green. Neste tipo de método – que é o caso do método implementado neste trabalho – a estimativa da imagem  $u^0|_D$  é feita apenas pela parte harmônica  $u^h$ , ou seja,  $u_D = u^h$ .

Por (5.18), o erro da estimativa em questão é exatamente a componente anti-harmônica  $u^a$ . Como  $u^0$  é uma função suave fixada, então existe uma constante positiva  $M$  tal que

$$|\Delta u^0(\mathbf{x})| \leq M$$

para todo  $\mathbf{x} \in D$ . Portanto, para qualquer  $\mathbf{x}_0 \in D$ , pelo Teorema (3.5.2),

$$|u(\mathbf{x}_0) - u^h(\mathbf{x}_0)| = |u^a(\mathbf{x}_0)| \leq M \int_D G(\mathbf{x}_0, \mathbf{x}) d\mathbf{x} \leq \frac{Md^2}{4}.$$

Conclui-se, portanto, que métodos que aproximam a região danificada apenas por extensão harmônica caracterizam-se por serem de ordem linear (c.f. CHAN; SHEN, 2002), o que significa que, para toda imagem  $u^0$  suave, quando  $d \rightarrow 0$ , ocorre

$$\|u_D - u^0\|_D = O(d^2)$$

que é o erro máximo que se comete no *inpainting* harmônico, considerando a hipótese de uma função  $u^0$  suave como imagem original.

Finalmente, é possível fazer duas considerações importantes sobre o método apresentado. Primeiramente, como se baseiam na convolução da imagem com um núcleo Gaussiano, a tendência é a suavização da imagem, gerando um efeito de borramento. Em segundo lugar, este é um método que vai funcionar com erro adequado quando a região danificada for suficientemente pequena, considerando seu caráter de erro quadrático. Logo, para grandes regiões o método também apresentará um erro suficientemente grande que faça a sua escolha ser descartada.

## 3.6 Filtragem

Os filtros podem ser entendidos como um conjunto de operações ou processos que será aplicado em um sinal para remover determinadas informações do seu espectro de frequências (GONZALEZ; WOODS, 2008). No contexto deste trabalho, o uso de alguns filtros em imagens serve tanto para atenuar efeitos ruidosos como também para recuperar

imagens danificadas. Muitos filtros são usados para remover ruídos de sinais analógicos e digitais, tornando a informação mais clara e limpa. Aqui, a remoção tem o mesmo objetivo: tornar a informação (no caso, os *pixels* que compõem a imagem) mais limpa.

Há duas formas de se aplicar um filtro em algum sinal e executar as operações desejadas – sejam elas remover imperfeições ou reduzir ruídos. De acordo com CARVALHO, 2003, pode-se dividir os filtros em dois tipos:

- filtros no domínio da frequência
- filtros no domínio do espaço

Os filtros de suavização de imagens servem para diminuir discontinuidades e diferenças entre um *pixel* e seu vizinho. Após o uso de um filtro de suavização, o resultado obtido no *pixel* que foi processado é a média entre ele e seus vizinhos, reduzindo, assim, os ruídos. No entanto, há o risco de afetar a imagem de forma que se altere muito a imagem real. Tais filtros dividem-se em filtros lineares e não lineares.

Os filtros lineares são aqueles que aplicam uma operação linear na imagem original e são passíveis de análise no domínio da frequência. Eles são utilizados em diversas áreas, não somente no processamento de imagem digital. Sua finalidade é eliminar frequências ou ruídos indesejáveis.

São exemplos de filtros lineares os filtros passa-baixa e passa-alta. Enquanto os primeiros suavizam a imagem atenuando as altas frequências (que correspondem às transições abruptas), os últimos atenuam as baixas frequências, produzindo um aguçamento (“*sharpening*”) da imagem, isto é, as transições entre regiões diferentes tornam-se mais nítidas. Os filtros passa-baixa tendem a minimizar ruídos e apresentam o efeito de borramento da imagem.

Diferentemente dos filtros lineares, os não lineares não possuem uma operação linear e também não são passíveis de análise no domínio da frequência. Esses tipos de filtros realçam ou minimizam ruídos por meio da variância. Neste caso, a média da imagem irá variar de acordo com cada filtro e a forma como são implementados.

Sistemas que utilizam operadores de detecção de borda são exemplos populares de uso dos filtros não lineares. *Softwares* baseados em operadores de detecção de borda também identificam manchas e curvas. Os operadores mais comuns são os operadores de Roberts e Sobel. Como resultado deles, obtém-se imagem com níveis de cinza para as bordas, manchas, curvas e linhas.

Outro exemplo de filtro não linear é o filtro morfológico. Esse tipo de filtro analisa as propriedades geométricas dos sinais (níveis de cinza da imagem). Os filtros morfológicos básicos são:

- Filtro morfológico da mediana: usado para suavização e remoção de ruído; mantém a dimensão da imagem
- Filtro morfológico de erosão: cria efeito de erosão nas partes claras da imagem (altos níveis de cinza), gerando imagens mais escuras.
- Filtro morfológico de dilatação: dilata as partes mais escuras da imagens (baixos níveis de cinza), gerando imagens mais claras.

É comum o uso encadeado dos dois últimos filtros para efeitos de abertura e fechamento de imagem que denotam, respectivamente, separação e união de elementos parecidos com ilhas, ligados a outras porções da imagem por pequenos traços.

### 3.6.1 Filtragem Digital

Uma imagem digital  $I$  é representada por uma ou mais matrizes de dimensão  $m \times n$ , a depender se a imagem é em tons de cinza ou colorida, respectivamente. O par  $(i, j)$  representa a posição do *pixel* na imagem  $I$  e o valor da matriz na posição  $(i, j)$  é o nível de cor naquele ponto. Uma imagem em preto e branco, portanto, é representada por apenas uma matriz com os níveis de cinza. Já uma imagem colorida possui três ou quatro matrizes correspondendo aos níveis de cor RGB ou CMYK, respectivamente.

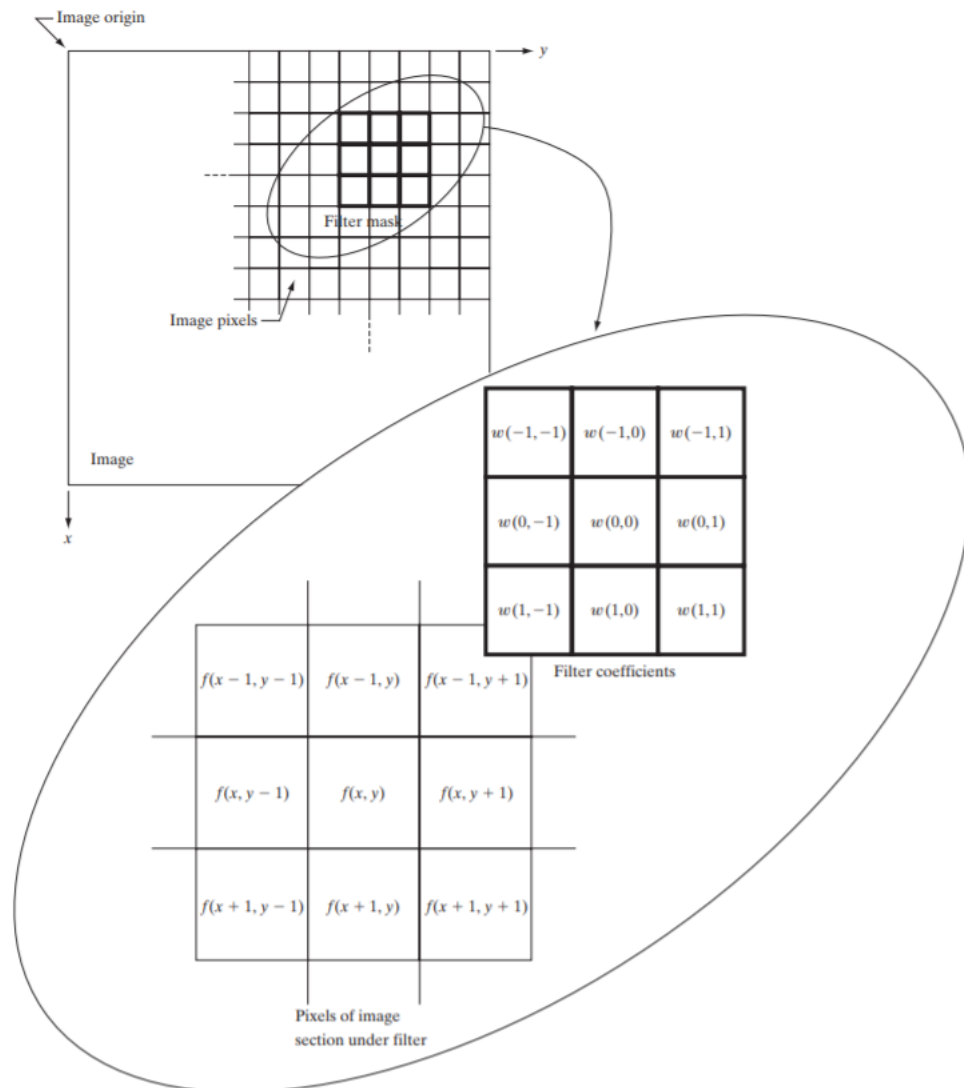
Conforme descrito no início desta seção, os filtros digitais para imagens podem agir no domínio da frequência ou do espaço. O algoritmo em R discutido neste trabalho realiza a filtragem no domínio do espaço. Diversos filtros digitais são baseados na convolução no domínio do espaço e são os operadores mais usados em processamento de imagens, com aplicações em pré-processamento, eliminação de ruídos, suavização e segmentação.

O processo de filtragem digital em imagens é similar à convolução de funções. Utilizam-se matrizes – as chamadas máscaras – a serem aplicadas sobre a imagem. A aplicação da máscara centrada na posição  $(i, j)$  da imagem, sendo  $i$  o número a linha e  $j$  o número da coluna de determinado *pixel*, consiste na substituição do valor do *pixel* nessa posição por um novo valor que depende dos valores dos *pixels* vizinhos e dos pesos da máscara, gerando uma nova imagem.

Conforme explicado por GONZALEZ; WOODS, 2008, o cálculo dos valores da imagem filtrada pode ser feito por meio da correlação ou da convolução da máscara com a imagem original. Na correlação, a máscara é deslocada, *pixel* a *pixel*, sobre a porção da imagem a ser reconstruída, calculando-se a soma dos produtos correspondentes de máscara e valor antigo em cada posição matricial. A convolução é o mesmo processo, com a diferença que a máscara é antes espelhada (rotacionada em  $180^\circ$ ).

A figura 3.6, de GONZALEZ; WOODS, 2008, ilustra a mecânica da filtragem digital espacial com o uso de uma máscara de dimensão  $3 \times 3$ . O ponto  $(x, y)$  da imagem

original é substituído pela média ponderada dos nove elementos usados no cálculo de  $g(x, y)$ : ele e seus oito vizinhos, cujo valor na imagem original é a imagem pela função  $f$ .



Fonte: GONZALEZ; WOODS, 2008

Figura 3.6 – Processo de filtragem espacial linear usando máscara  $3 \times 3$

Se a máscara tem dimensão ímpar  $m \times n$ ,  $m = 2a + 1$  e  $n = 2b + 1$ , com  $a$  e  $b$  inteiros positivos, então o coeficiente central da máscara  $w(0, 0)$  corresponde ao elemento  $(x, y)$  para o qual está sendo calculado o novo valor de luminosidade. Para uma máscara de tamanho par, apesar de não ser muito comum, a convenção é calcular o resultado sobre o primeiro elemento (ao invés do elemento central).

Vale lembrar que a imagem tem apenas uma matriz se for em níveis de cinza, mas tem três ou quatro matrizes se for colorida, cada matriz para as cores RGB ou CMYK, respectivamente. Nestes dois últimos casos, o processo é feito para cada componente de cor separadamente.

As equações abaixo representam o cálculo para a filtragem espacial utilizando correlação e convolução, denotadas por  $\diamond$  e  $*$  respectivamente, com máscaras de dimensão  $m \times n$ ,  $m = 2a + 1$  e  $n = 2b + 1$ , com  $a$  e  $b$  inteiros positivos. Nas equações,  $(x, y)$  percorre cada *pixel* da porção da imagem a ser aplicado o processo de remoção de ruído e o *inpainting* na imagem  $f$ .

$$\text{Correlação: } w(x, y) \diamond f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

$$\text{Convolução: } w(x, y) * f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x - s, y - t).$$

Como resultado, se a máscara for simétrica, o resultado da correlação e da convolução serão iguais. Este é o caso da maioria das técnicas de processamento digital de imagens.

O custo computacional da convolução espacial pode ser bastante alto. Uma imagem de tamanho  $m \times m$  e máscara  $n \times n$  precisará de  $m^2 n^2$  operações, entre multiplicações e somas, para que seja executada uma convolução espacial. Uma alternativa é utilizar a filtragem digital no domínio da frequência com transformada de Fourier, principalmente para máscaras muito grandes. Neste tipo de filtragem, o primeiro passo é transformar a imagem do domínio espacial para o domínio da frequência, por meio da transformada de Fourier bidimensional

$$\mathcal{F}(f(\mathbf{x})) = F(\mathbf{u}) = \frac{1}{2\pi} \int_{\mathbb{R}^2} f(\mathbf{x}) e^{-2i\pi \mathbf{x} \cdot \mathbf{u}} d\mathbf{x},$$

em que  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ ,  $f \in L^1(\mathbb{R}^2)$ ,  $\mathbf{x} = (x, y)$  e  $\mathbf{u} = (u, v)$ .

Considerando que a filtragem se dá por um processo convolucional, e no domínio da frequência a convolução é transformada em uma multiplicação entre transformada do filtro  $w$  pela transformada da imagem  $f$ , isto é,  $\mathcal{F}\{w * f\} = 2\pi \mathcal{F}\{w\} \cdot \mathcal{F}\{f\}$ , então é, em muitos casos, vantajoso trabalhar no domínio da frequência ao invés do espacial (GONZALEZ; WOODS, 2008). O último passo é aplicar a transformada de Fourier inversa para obter o resultado final

$$\mathcal{F}^{-1}(F(\mathbf{u})) = f(\mathbf{x}) = \frac{1}{2\pi} \int_{\mathbb{R}^2} F(\mathbf{u}) e^{2i\pi \mathbf{x} \cdot \mathbf{u}} d\mathbf{u}.$$

No caso da filtragem digital, é usada uma versão discreta da transformada de Fourier e sua inversa. Seja  $f(x, y)$  uma imagem de tamanho  $m \times x$ , então a transformada rápida e sua inversa são calculadas da forma:

$$\mathcal{F}(f(x, y)) = F(u, v) = \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f(x, y) e^{-2i\pi \left( \frac{u \cdot x}{m} + \frac{v \cdot y}{n} \right)}$$

$$\mathcal{F}^{-1}(F(u, v)) = f(x, y) = \frac{1}{m n} \sum_{u=0}^{m-1} \sum_{v=0}^{n-1} F(u, v) e^{2i\pi \left( \frac{u \cdot x}{m} + \frac{v \cdot y}{n} \right)}$$

para  $x, u \in 0, 1, 2, \dots, m - 1$  e  $y, v \in 0, 1, 2, \dots, n - 1$ .

No domínio da frequência, a filtragem pode ser efetuada em tempo  $O((mn) \ln(mn))$  utilizando a transformada rápida de Fourier. Além disso, a operação de convolução, e consequentemente filtragem, é drasticamente simplificada no domínio da frequência, considerando as propriedades de uma transformada integral sobre a convolução, visto que esta se reduz a uma multiplicação das transformadas.

Um detalhe importante na implementação da filtragem digital é como tratar a borda da imagem. No processo de correlação ou convolução da máscara pela imagem, conforme ilustrado na figura 3.6, quando o *pixel* central está na borda ou perto dela, perde-se informação de muitos *pixels* em torno dele, já que estariam fora da imagem original. Para tratar esse problema, algumas alternativas são: preencher a imagem com valores padrão (como branco, por exemplo), ignorar a borda (resultando em uma imagem menor que a original) ou repetir para fora da imagem a informação da borda (ou de seu interior), em dimensão suficiente para que o processo funcione em toda a imagem.

O algoritmo implementado em R replica os dados da borda para fora de modo que possa ser reconstruída a imagem por completo, incluindo a borda. Este tratamento é um detalhe importante e, muitas vezes, esquecido nas implementações iniciais, sendo lembrado apenas quando o erro ocorre visualmente.

### 3.6.2 Filtro Gaussiano

Um filtro passa-baixa comum nos estudos que envolvem recuperação de imagens é o filtro Gaussiano. Ele é útil em diferentes áreas de processamento de imagens. Algumas dessas diferentes utilidades foram descritas por PEDRINI; SCHWARTZ, 2007:

- O filtro Gaussiano em duas dimensões tem o grau de suavização igual em ambas as direções, pois funções Gaussianas são simétricas em relação à rotação. Essa propriedade é chamada de isotropia.
- Neste filtro, assim como em outros, a suavização da imagem é realizada através da substituição de cada *pixel* pela média ponderada dos *pixels* vizinhos. Porém, o filtro Gaussiano funciona de forma que o peso dado a um vizinho decresce monotonicamente com a sua distância ao *pixel* central.
- O grau de suavização deste filtro depende do parâmetro  $\sigma$ , que está diretamente ligado à dimensão da matriz (máscara), ou seja, quanto maior a sua dimensão, maior a largura do filtro e maior o grau de suavização que será aplicado por ele.

- Funções Gaussianas são separáveis, portanto, é possível aplicar o filtro Gaussiano unidirecional em uma direção e, depois, aplicar sobre o resultado obtido o mesmo filtro na outra direção, resultando em uma convolução Gaussiana. Portanto, o custo computacional da utilização de um filtro Gaussiano 2D é linearmente proporcional ao tamanho da máscara usada, ao invés de crescer com o seu quadrado.

O filtro Gaussiano tem esse nome por usar derivações da função Gaussiana bidimensional de média zero para obter os coeficientes da máscara que é representada por matrizes de ordem ímpar (por motivos de simetria). Esta função é definida como

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

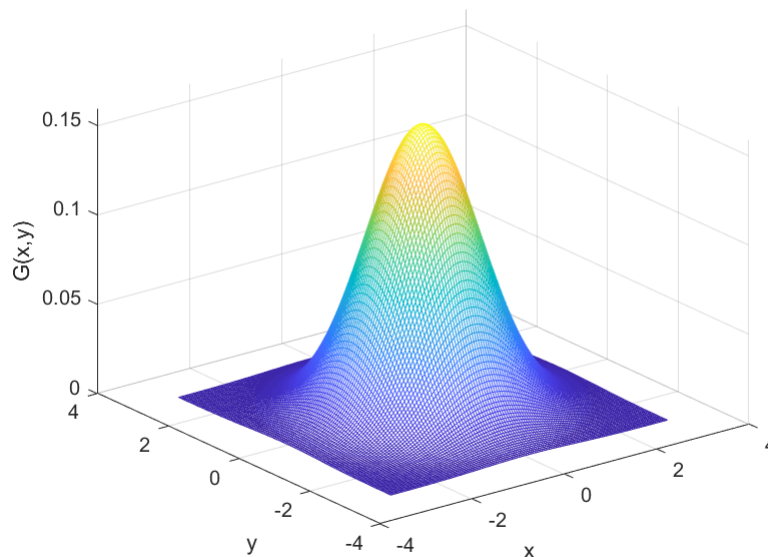


Figura 3.7 – Função Gaussiana de duas variáveis com  $\sigma = 1$

Na equação, o desvio-padrão é representado por  $\sigma$ . Ele é o parâmetro que determina a largura do filtro. Já os valores de  $x$  e  $y$  representam as distâncias dos *pixels* a serem ponderados até o *pixel* central e, conseqüentemente, definem o tamanho da matriz que representa a máscara. Por exemplo, podem ser usados  $x$  e  $y$  variando de  $-2$  a  $2$ , gerando a matriz que representa a máscara de dimensão  $5 \times 5$ .

Para a construção da máscara, deve-se utilizar a última das propriedades listadas acima: gerar  $G(x)$ ,  $G(y)$  e multiplicá-las. Um detalhe é que  $G(x)$  e  $G(y)$  são vetores-coluna, logo, a multiplicação deve ser  $G(x, y) = [G(x)] \cdot [G(y)]^T$ . A função Gaussiana de uma variável é definida como

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

É comum o uso de máscaras  $5 \times 5$  pois engloba 98,8% da distribuição Gaussiana com  $\sigma = 1$ . No caso da máscara de dimensão  $5 \times 5$ , os valores inteiros  $x$  e  $y$  variam de  $-2$  a  $2$ . Com isso, é possível calcular os vetores  $G(x)$  e  $G(y)$  para o caso em que  $\sigma = 1$ .

$$\begin{aligned} G(-2) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{(-2)^2}{2}} \approx 0,0540 \\ G(-1) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{(-1)^2}{2}} \approx 0,2420 \\ G(0) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{(0)^2}{2}} \approx 0,3989 \\ G(1) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{(1)^2}{2}} \approx 0,2420 \\ G(2) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{(2)^2}{2}} \approx 0,0540 \end{aligned}$$

Como os números não são inteiros, multiplica-se (e depois divide-se) todos eles por  $2^{n-1}$ , onde  $n$  é a ordem da máscara (neste caso,  $n = 5$ ), arredondando-os. Desta forma, os filtros são, em ordem, 1, 4, 6, 4 e 1. É possível obter os coeficientes da máscara Gaussiana de ordem  $n$  por meio da  $n^{\text{a}}$  linha do Triângulo de Pascal, com  $\sigma = \frac{\sqrt{n-1}}{2}$ . Isto segue do fato de que a distribuição Gaussiana (ou normal) aproxima-se da distribuição binomial, conforme o Teorema abaixo demonstrado em BOAS, 2006.

**Teorema 3.6.1.** *Seja  $x$  uma variável aleatória de distribuição binomial  $B(n, p)$ . Então, para  $n$  suficientemente grande, a seguinte variável aleatória segue uma distribuição normal padrão:*

$$z = \frac{x - \mu}{\sigma} \sim N(0, 1),$$

onde

$$\mu = np, \sigma^2 = np(1 - p).$$

Com isso, obtém-se

$$G(x, y) = \frac{1}{16} \begin{pmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{pmatrix} \odot \frac{1}{16} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \end{pmatrix} = \frac{1}{256} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix},$$

em que  $\odot$  representa a multiplicação ponto a ponto.

Esta é uma das matrizes de máscara  $5 \times 5$  para o filtro Gaussiano bidimensional. Há na literatura, como em FARIA, 2005 e SANCHES et al., 2015, a matriz em que o



denominador é 273 ao invés de 256. O mais importante é que esteja normalizada, ou seja, a soma dos elementos da matriz é igual ao denominador dos elementos. O objetivo é escrever os números obtidos pela função Gaussiana como frações de denominador igual à soma de todos os numeradores.

A matriz  $5 \times 5$  do filtro Gaussiano exibida em FARIA, 2005 e SANCHES et al., 2015 é obtida apenas dividindo-se cada termo em  $G(x, y)$  pelo menor de todos, arredondando-se em seguida:

$$G(x, y) = \frac{1}{273} \begin{pmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{pmatrix}$$

### 3.7 Implementação em R

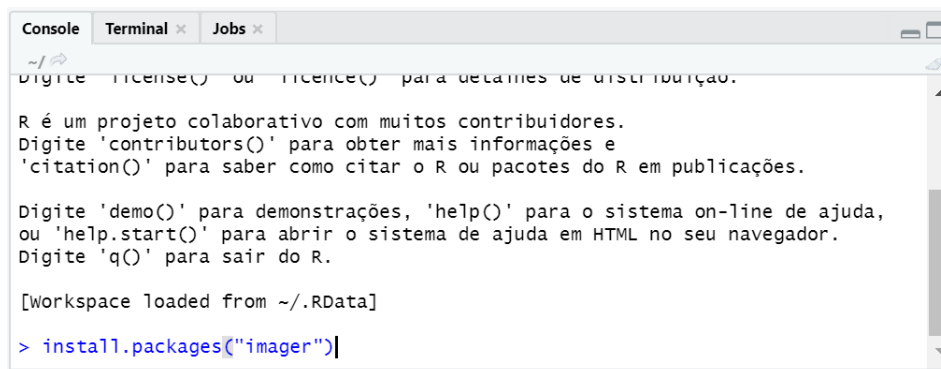
Muitas são as técnicas utilizadas no âmbito do *inpainting*. A biblioteca **imager** desenvolvida na linguagem de programação R foi a escolhida para se trabalhar o assunto com os alunos do Ensino Médio por ser de fácil compreensão e possuir funções simples de manipulação de imagens.

Além disso, a **imager** é baseada em uma biblioteca bastante utilizada em Processamento de Imagens, que é a CImg, desenvolvida por David Tschumperle em C++.

A documentação do pacote **imager** pode ser obtida em <https://rdr.io/cran/imager/>. Algumas de suas funções são:

- **grayscale**: coloca a imagem em escala de cinza conforme o método escolhido
- **imfill**: cria uma imagem de tamanho escolhido repetindo valores de entrada
- **imresize**: ajusta a imagem para o tamanho do parâmetro de entrada
- **imrotate**: rotaciona a imagem dado o ângulo e o centro da rotação
- **imwarp**: faz um remapeamento dos valores dos *pixels*
- **nPix**: tem como saída o número de *pixels* da imagem
- **inpaint**: aplica o filtro Gaussiano na suavização de imagens, removendo ruídos

A sua instalação se dá por meio do comando **install.packages(imager)** executado no Console do RStudio, conforme tela abaixo.



```
~/ |
Digite 'license()' ou 'licence()' para detalhes de distribuição.

R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publicações.

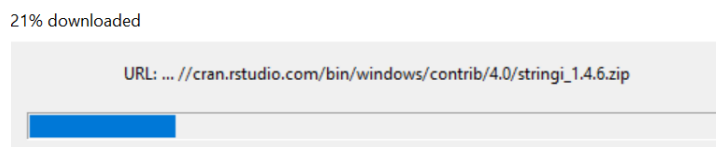
Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de ajuda,
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu navegador.
Digite 'q()' para sair do R.

[Workspace loaded from ~/.RData]

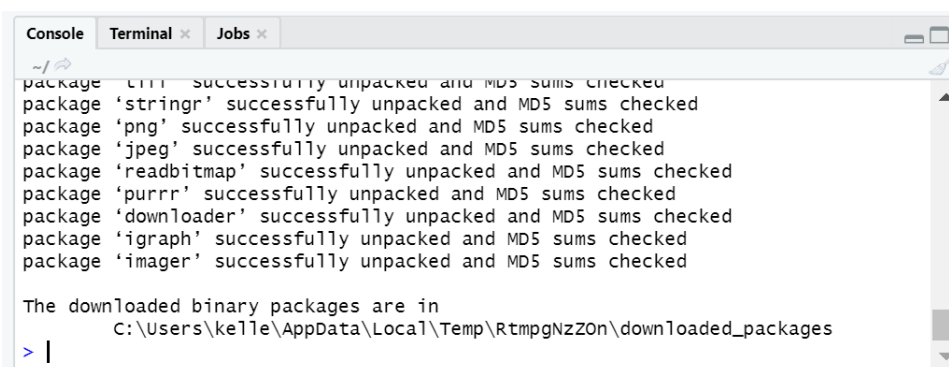
> install.packages("imager")
```

Figura 3.8 – Comando no Console para instalação da biblioteca **imager**

Ao executar o comando, o *software* instala os pacotes necessários para a completa instalação da biblioteca, com algumas janelas de evolução conforme a da figura abaixo.

Figura 3.9 – Janela de progresso de *download* de pacotes que a biblioteca **imager** utiliza

Por fim, após tudo carregado, no Console do RStudio é exibido o diretório onde ficou salva a biblioteca, estando pronto para receber os próximos comandos.



```
~/ |
package 'litt' successfully unpacked and MD5 sums checked
package 'stringr' successfully unpacked and MD5 sums checked
package 'png' successfully unpacked and MD5 sums checked
package 'jpeg' successfully unpacked and MD5 sums checked
package 'readbitmap' successfully unpacked and MD5 sums checked
package 'purrr' successfully unpacked and MD5 sums checked
package 'downloader' successfully unpacked and MD5 sums checked
package 'igraph' successfully unpacked and MD5 sums checked
package 'imager' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\kelle\AppData\Local\Temp\RtmpgNZZ0n\downloaded_packages

> |
```

Figura 3.10 – Console após a completa instalação da biblioteca **imager**

O exemplo a ser trabalhado com os alunos consiste em um programa que insere alguns *bits* vazios (com valor “NA”), ou seja, sem informação, o que, na prática, equivale a retirar as informações. Após isso, a função **inpaint** da biblioteca realiza a reconstrução da imagem danificada por meio do filtro Gaussiano, isto é, estimar e preencher os *pixels* vazios como uma média ponderada dos *pixels* vizinhos. A imagem sugerida para o exercício é a mesma do exemplo na documentação oficial do pacote **imager** e todas as funções utilizadas são desse pacote.

Figura 3.11 – Imagem *boats*

Com a biblioteca instalada, o próximo passo é carregá-la no R com o uso do comando **library(imager)** no código-fonte de extensão **.r** ou na linha de comando do Console. Este comando representa a primeira linha do código-fonte sugerido para se trabalhar em sala de aula, apresentado a seguir:

```
1 library(imager)
  im <- boats
3 im[sample(nPix(im), 1e4)] <- NA
  novaIm = inpaint(im, 1)
5 lista = imlist(im, novaIm)
  lista = setNames(lista, c("Antes", "Depois"))
7 plot(lista, layout="row")
```

A segunda linha do código faz com que a variável **im** guarde a imagem *boats* a ser trabalhada no exercício. Aqui vale uma observação importante: é necessário que a imagem com o nome **boats.png** esteja salva na mesma pasta que o código **.r** acima descrito para que este funcione como está. Caso a imagem esteja salva em outro diretório, é preciso descrever na segunda linha o caminho completo ou incluir o caminho da pasta da imagem no *path* do R.

Para a inserção aleatória de ruídos pela imagem, o comando é o da linha 3. Inicialmente, cria-se uma amostra de dez mil valores (ou **1e4**) entre 1 e o número de *pixels* da imagem (obtido pela função **nPix**), inclusive. Com esses dez mil valores, faz-se a atribuição de “NA” dos *pixels* numerados conforme os números da amostra. Ou seja, agora a

imagem dos barcos é idêntica à original, porém com dez mil *pixels* em branco, resultado do valor “NA” na cor de suas respectivas posições na imagem.

A quarta linha do código-fonte faz uso da função **inpaint** para reconstruir a imagem. Essa função utiliza o filtro Gaussiano na reconstrução da imagem, ou seja, na retirada dos ruídos criados no passo anterior. Esse procedimento consiste em substituir cada *pixel* que está faltando por uma média ponderada dos seus vizinhos. A função **inpaint** recebe dois argumentos. O primeiro é a imagem a ser reconstruída e o segundo é o desvio-padrão do filtro. Neste caso, foi usado o desvio-padrão 1. O resultado dessa operação é armazenado na variável **novaIm**.

Após isso, na linha 5, é criada uma variável **lista** que apenas contém as duas imagens: com ruídos e reconstruída. Na linha 6, renomeiam-se as duas imagens para “Antes” e “Depois”, respectivamente. Por fim, a última linha apenas exhibe as duas figuras lado a lado.

O resultado é exibido na aba Plots do RStudio. A comparação visual entre a figura original (figura 3.11) e a figura obtida após a aplicação do método de *inpainting* (figura 3.12) é interessante e mostra que o método é eficaz.

Em termos de precisão e eficácia do método, é possível fazer o exercício para porções maiores de valores “NA”. Para isso, basta substituir o comando

```
im [sample(nPix(im),1e4)] <- NA
```

para preencher com valores vazios alguns traços ou blocos.

```
1 'tracos'
  for(i in 140000:141000){
3     im[i] <- NA
  }
5 for(i in 153000:154000){
  im[i] <- NA
7 }
  for(i in 160000:161000){
9     im[i] <- NA
  }
11
12 'bloco'
13 for(i in 140000:150000){
  im[i] <- NA
15 }
16
17 'bloco mais largo'
  for(i in 100000:150000){
19     im[i] <- NA
  }
```

Os resultados estão a seguir. É interessante notar que o método, mesmo sendo um dos mais simples (via extensão harmônica), é satisfatório para falhas em *pixels* espaçados, bem como para pequenos traços de falha, como nas figuras 3.12 e 3.13. Como esperado, falhas em blocos tendem a ficar com um resultado pior, porém, devido à simplicidade do método, considera-se bom o que ele faz para os casos das figuras 3.14 e 3.15.

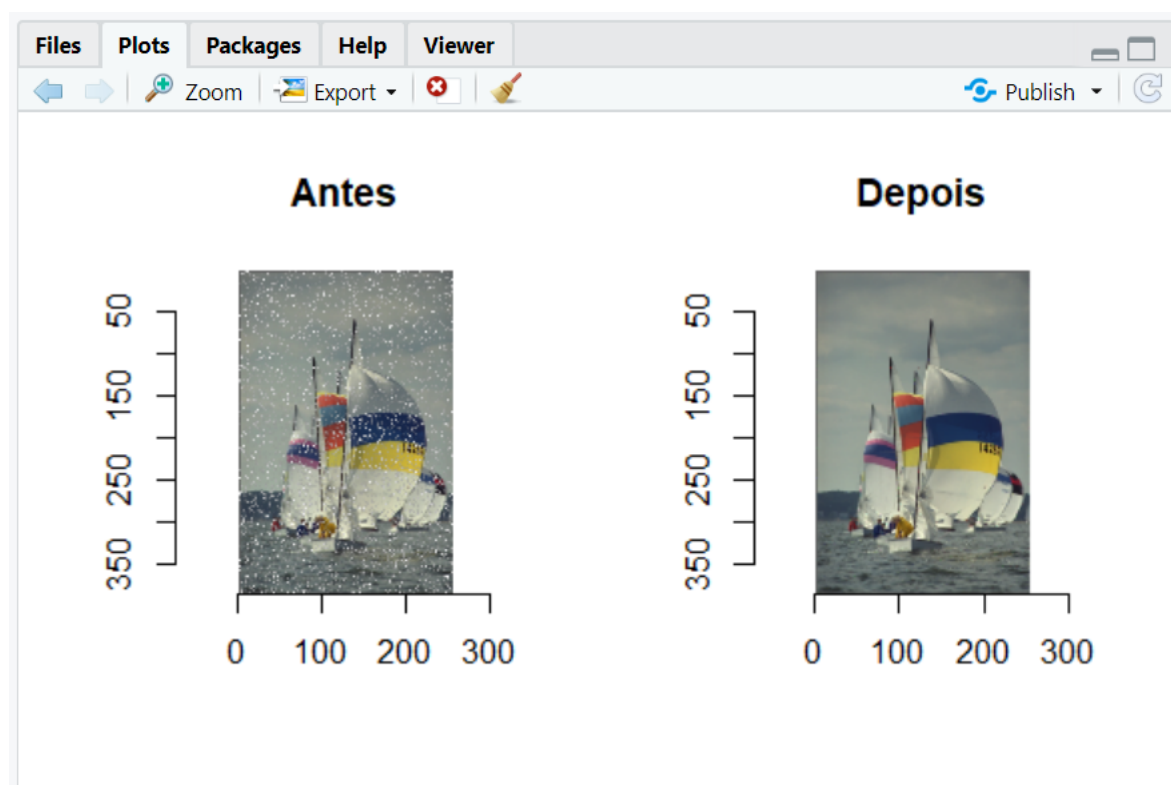


Figura 3.12 – Resultado da aplicação do filtro Gaussiano na imagem *boats* com falhas espalhadas

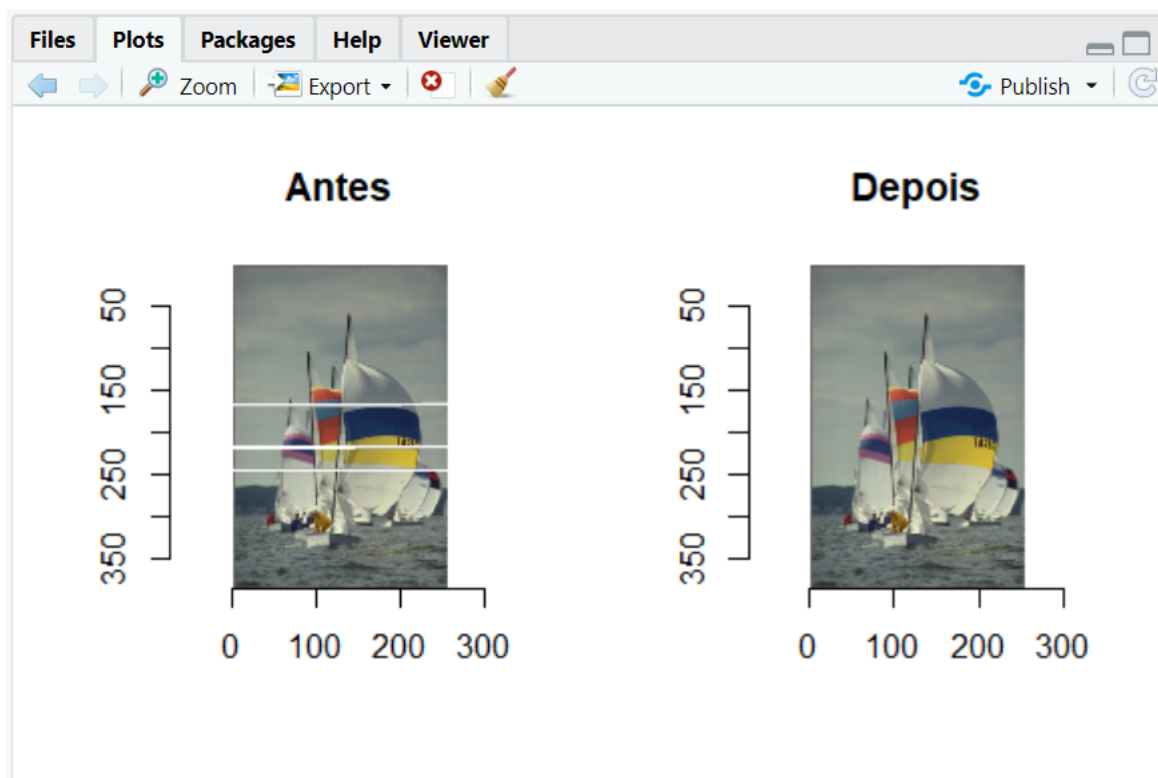


Figura 3.13 – Resultado da aplicação do filtro Gaussiano na imagem *boats* com falhas em traços

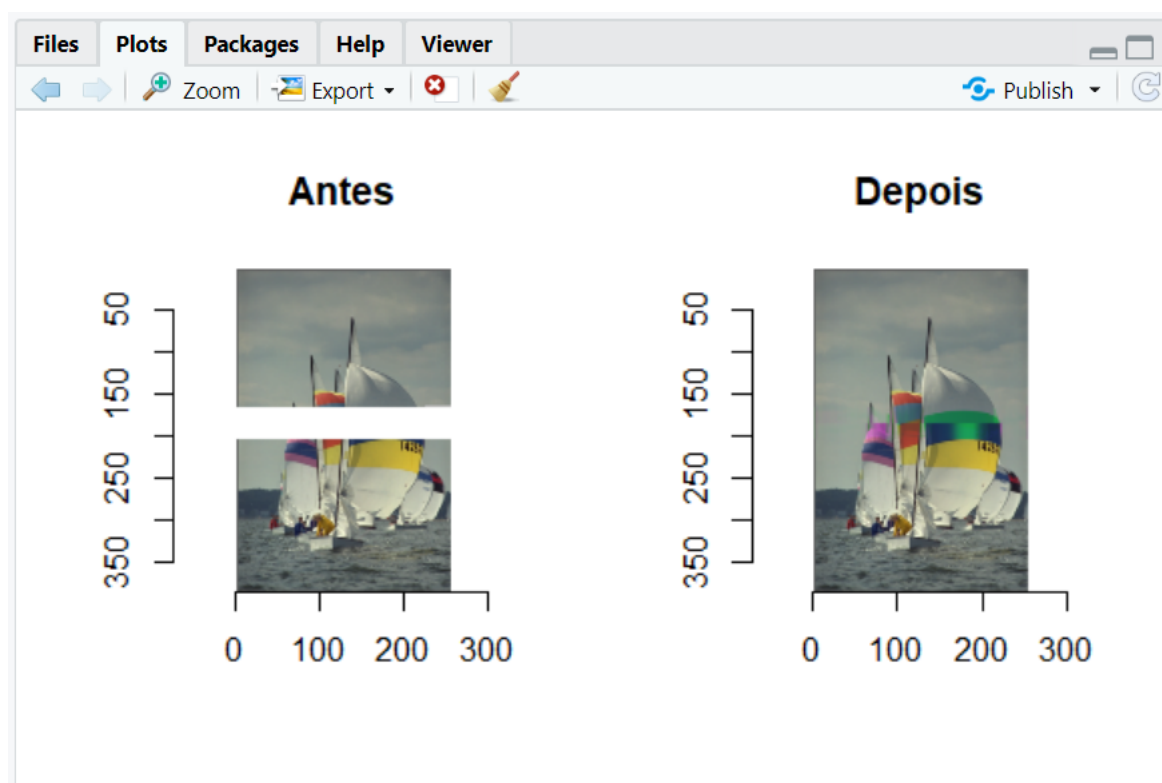


Figura 3.14 – Resultado da aplicação do filtro Gaussiano na imagem *boats* com falha em bloco

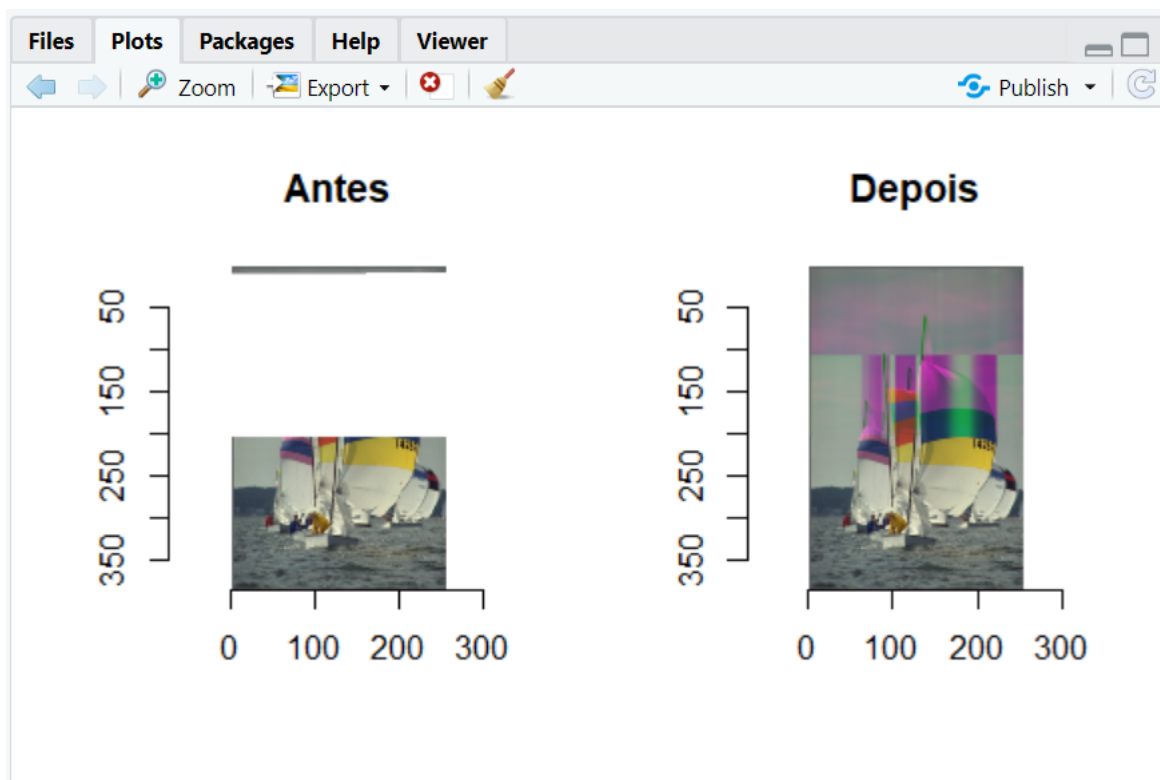


Figura 3.15 – Resultado da aplicação do filtro Gaussiano na imagem *boats* com falha em bloco mais largo

## 4 Roteiros de Aula

Este capítulo contém os roteiros de aula para o ensino da linguagem de programação R e da técnica de *inpainting*. Também são incluídos nos roteiros presentes nesse capítulo recursos requeridos, metodologia de ensino e, quando for o caso, sugestão de avaliação. Nele há cada roteiro com o tempo sugerido de aula e o passo a passo para aplicação em sala de aula.

O objetivo das aulas sugeridas neste capítulo é a implementação, pelos alunos em sala, de um programa de reconstrução de imagens em R. Esse programa recebe uma foto de entrada, insere algumas faltas de *bits* nela e, posteriormente, usando a biblioteca de *inpainting*, a imagem é reconstruída. Também são incluídas neste capítulo estrutura requerida, alternativas e metodologia de ensino. Nele há cada roteiro com o tempo sugerido de aula e o passo a passo para aplicação em sala de aula.

Para isso, sugere-se a aplicação de 13 aulas de 50 minutos, a serem aplicadas, preferencialmente, duas vezes na semana, totalizando 6 semanas e meia, ou perto de um mês e meio, para o projeto ser concluído.

Os roteiros de aula para alunos do Ensino Médio incluem, nesta ordem:

1. Breve ensino da linguagem R
2. Exemplos de aplicações em R
3. Inserção de falhas na imagem
4. Breve explanação sobre filtro Gaussiano para reconstrução de imagens
5. Remoção das falhas da imagem

### 4.1 Breve Ensino da Linguagem R

Apesar de não ser disciplina obrigatória na grade curricular do Ensino Médio, a programação tem aparecido cada vez mais nas aulas.

O objetivo desta primeira seção é abordar a linguagem de programação R com alunos do Ensino Médio. Para isto, são sugeridas quatro aulas de 50 minutos cada uma, abordando os respectivos conteúdos:

- **Aula 1** : A linguagem R e o RStudio
- **Aula 2** : Variáveis e Funções



- **Aula 3** : Uso de Bibliotecas

#### 4.1.1 Aula 1: A linguagem R e o RStudio

Aula 1	
Professor:	
Disciplina:	
Tema:	A linguagem R e o RStudio
Turma:	
Data:	
Duração da aula:	50 minutos

### Objetivos

Esta aula apresenta a linguagem de programação R e o *software* RStudio que serve de compilador e ambiente de programação. É uma aula em que o professor apresenta aos alunos a linguagem R, com suas peculiaridades e principais usos (que começaram na Estatística), conforme consta na seção 2.1. Para melhor rendimento da aula, é recomendado o uso do RStudio projetado em sala para que os alunos possam ver a execução dos comandos.

Recomenda-se iniciar a aula exibindo-se o antes e o depois da imagem com as falhas e recuperada, respectivamente, para mostrar aos alunos a motivação das aulas. Como exemplos, podem ser exibidas as figuras 3.1, 3.2, 3.3 e 3.4.

Também é interessante mostrar alguns exemplos simples de utilização do RStudio, ainda sem uso de variáveis ou funções. Por exemplo, mostrando que ele exibe o valor de  $\pi$  e também faz contas aritméticas respeitando, claro, a hierarquia de procedência entre as operações. Pode-se passar algumas contas longas com muitas operações e pedir que eles calculem de cabeça, validando o resultado no RStudio ao executar o comando.

Espera-se que os alunos, após esta aula, tenham condições de entender o contexto em que o R foi criado, saber a diferença entre R e RStudio, bem como saber lidar com funções básicas do menu do RStudio.

### Conteúdos

- Definição do R
- Contexto histórico da criação do R

- Apresentação do RStudio
- Exemplos de comandos no R por meio do RStudio
- Dinâmica com os alunos com contas

## Procedimentos metodológicos

Apresentação interativa do conteúdo por meio da projeção da tela do computador com R e RStudio instalados.

## Recursos didáticos

- Computador com R e RStudio instalados
- Projetor

### 4.1.2 Aula 2: Variáveis e Funções

Aula 2	
Professor:	
Disciplina:	
Tema:	Variáveis e Funções
Turma:	
Data:	
Duração da aula:	50 minutos

## Objetivos

Esta aula – que pode ser feita já em laboratório, se houver a possibilidade – aborda o estudo de variáveis e funções no R, de forma lúdica e não muito aprofundada, tendo em vista o público-alvo. O professor pode exemplificar com algumas atribuições simples utilizando as duas formas permitidas na sintaxe do R, como `a = 2` e `a <- 2`. Também é nesta aula que o professor pode mostrar que os comandos de atribuição de valores a variáveis podem ser executados diretamente na linha de comando como também podem ser listados no arquivo de extensão `.r` para posterior execução do arquivo inteiro. Para maiores informações, é possível consultar a seção 2.2.

Para apresentar as funções, pode-se implementar a função fatorial e depois fazer chamadas em arquivo `.r` e também na linha de comando.

Após esta aula, os alunos devem ter condições de criar funções e variáveis em R utilizando o RStudio.

## Conteúdos

- Definição de variáveis e funções no contexto de programação
- Criação de variáveis no R
- Atribuição de valores a variáveis no R
- Definição de funções no R
- Chamada de funções no R
- Exercício para os alunos implementarem alguma função simples e chamar para fazer um cálculo com o seu auxílio

## Procedimentos metodológicos

Aula em laboratório de informática interativa, com apresentação do conteúdo por meio da projeção da tela do computador com R e RStudio instalados.

## Recursos didáticos

- Computador com R e RStudio instalados
- Projetor
- Laboratório de informática cujos computadores possuam R e RStudio instalados

### 4.1.3 Aula 3: Uso de Bibliotecas

Aula 3	
Professor:	
Disciplina:	
Tema:	Uso de Bibliotecas
Turma:	
Data:	
Duração da aula:	50 minutos

## Objetivos

Esta aula tem por objetivo ensinar aos alunos como se instala e utiliza bibliotecas no R. O professor pode explicar que o R já conta com vários pacotes já instalados e mostrar exemplos de uso das mesmas em códigos de extensão `.r`, com exemplos, conforme seção 2.2.

Com isso, o professor inicia a aula definindo o que vem a ser uma biblioteca – pacote de funções. Então, faz uma chamada a uma já instalada como padrão para, posteriormente, exibir exemplos de instalação e uso de bibliotecas novas. É importante passar um exercício em sala de forma que os alunos pratiquem.

Após esta aula, os alunos devem ter condições de instalar pacotes no R, bem como de fazer chamadas de funções de um determinado pacote instalado, tudo por meio da linha de comando e do RStudio.

## Conteúdos

- Definição de bibliotecas
- Diferenciação de bibliotecas padrão para as novas (citar que o R é uma linguagem *open source* e qualquer pessoa pode disponibilizar um pacote na internet)
- Utilização de funções de bibliotecas já instaladas no R
- Instalação de biblioteca nova
- Chamada de funções da biblioteca nova que foi instalada
- Exercício para os alunos instalarem e implementarem um código que use função da nova biblioteca

## Procedimentos metodológicos

Aula em laboratório de informática interativa, com apresentação do conteúdo por meio da projeção da tela do computador com R e RStudio instalados.

## Recursos didáticos

- Computador com R e RStudio instalados
- Projetor
- Laboratório de informática cujos computadores possuam R e RStudio instalados

## 4.2 Exemplos de Aplicações em R

Após as aulas anteriores, os alunos já terão condições de implementar alguns exemplos menos triviais em R. Nestes exemplos de aplicações, pode-se escolher algum tema interessante para trabalhar o desenvolvimento de um código em sala. É interessante, para encaminhar o processo de *inpainting* em sala, utilizar funções de manipulação de imagens.

O objetivo desta seção é realizar alguns exemplos em R com os alunos e deixá-los implementar soluções de problemas simples na linguagem R. Para isto, são sugeridas três aulas de 50 minutos cada uma, abordando os respectivos conteúdos:

- **Aula 4** : Cálculo Aritmético em R
- **Aula 5** : Recorrências em R
- **Aula 6** : Manipulação de Arquivos e Imagens em R

### 4.2.1 Aula 4: Cálculo Aritmético em R

Aula 4	
Professor:	
Disciplina:	
Tema:	Cálculo Aritmético em R
Turma:	
Data:	
Duração da aula:	50 minutos

#### Objetivos

Esta aula tem por objetivo trabalhar com os alunos exemplos de código em R para executar cálculos aritméticos. Pode-se passar dois exercícios similares e dar cerca de 25 minutos para os alunos, individualmente ou em grupo, resolvê-los em computador. No restante do tempo, segue debate sobre dificuldades e ideias que surgiram no desenvolvimento dos exercícios, finalizando com o professor exibindo uma forma de resolver os problemas. Também é possível pedir para os alunos explicarem os seus códigos, fomentando a participação e a interação na turma.

Com isso, espera-se que os alunos tenham condições de programar e executar, no RStudio, códigos para receber parâmetros de entrada digitados pelo usuário, com menu de opções de operações ariméticas simples, devolvendo o resultado na tela.

## Conteúdos

- Exercício: desenvolver um programa em R tal que o usuário digite dois números e o programa devolva sua soma, subtração, multiplicação e divisão
- Exercício: desenvolver um programa em R que exiba um menu com as quatro opções básicas de operação aritmética; o usuário escolhe uma delas e insere dois números; o resultado é a operação entre os dois números digitados

## Procedimentos metodológicos

Aula em laboratório com programação feita pelos alunos, individualmente ou em grupo.

## Recursos didáticos

- Computador com R e RStudio instalados
- Projetor
- Laboratório de informática cujos computadores possuam R e RStudio instalados

## Avaliação

A avaliação, se desejada, pode ser feita por meio da entrega ou da apresentação para a turma (caso em que será necessário mais do que 50 minutos de aula) do código-fonte que cada aluno ou grupo obteve.

### 4.2.2 Aula 5: Recorrências em R

Aula 5	
Professor:	
Disciplina:	
Tema:	Recorrências em R
Turma:	
Data:	
Duração da aula:	50 minutos

## Objetivos

Esta aula aborda com os alunos recorrências em programação. O professor pode começar exemplificando, nos primeiros 10 minutos de aula, como calcular o fatorial de um número (programando o caso  $0!$  à parte) de forma recursiva. Então, o desafio é deixar os alunos tentarem desenvolver, por cerca de 25 minutos, com a tutoria do professor em sala que pode auxiliar retirando dúvidas, algum problema de recorrência por meio da programação na linguagem R. O final da aula, assim como a aula anterior, serve de debate das soluções.

O exemplo das Torres de Hanói é clássico e pode-se trabalhar também, se o professor quiser, o método para encontrar a fórmula geral analiticamente, de forma que o programa implementado sirva de validação.

Com esta aula, é esperado que os alunos sejam capazes de programar no RStudio um código que, recebendo valores de entrada ou apenas usando um número fixo de elementos, calcule o número de movimentos para o problema das Torres de Hanói. Uma variação mais simples, a depender do nível da turma, é calcular o  $n$ -ésimo número da sequência de Fibonacci.

## Conteúdos

- Exercício: desenvolver um programa em R que resolva um problema de recorrência (como Torres de Hanói ou Fibonacci)

## Procedimentos metodológicos

Aula em laboratório com programação feita pelos alunos, individualmente ou em grupo.

## Recursos didáticos

- Computador com R e RStudio instalados
- Projetor
- Laboratório de informática cujos computadores possuam R e RStudio instalados

## Avaliação

A avaliação, se desejada, pode ser feita por meio da entrega ou da apresentação para a turma (caso em que será necessário mais do que 50 minutos de aula) do código-fonte que cada aluno ou grupo obteve.

### 4.2.3 Aula 6: Manipulação de Arquivos e Imagens em R

Aula 6	
Professor:	
Disciplina:	
Tema:	Manipulação de Arquivos e Imagens em R
Turma:	
Data:	
Duração da aula:	50 minutos

## Objetivos

Os objetivos desta aula são ensinar como se manipula arquivos e imagens na linguagem R, de forma breve e simples.

A aula é dividida em duas partes: os primeiros 30 minutos são de exposição de *slides* e exemplos de códigos no RStudio; a segunda parte (20 minutos restantes) são deixadas para os alunos, individualmente ou em grupo, fazerem exercícios de treinamento com leitura e gravação de arquivos, bem como manipulação básica de imagens.

Desta forma, espera-se que os alunos sejam capazes de ler, deletar e salvar um arquivo via comandos em R, bem como de carregar uma imagem e obter algumas propriedades da imagem carregada, como tamanho, local da máquina em que está salva, etc.

## Conteúdos

- Definição de arquivo em R
- Sintaxe das principais operações com arquivos em R
- Manipulação de imagens em R
- Exercícios de treinamento com arquivos e imagens em R



## Procedimentos metodológicos

Aula expositiva e interativa em laboratório de informática, com apresentação de *slides* e da tela projetada do RStudio e com exercícios feitos pelos alunos nos computadores.

## Recursos didáticos

- Computador com R e RStudio instalados
- Projetor
- Laboratório de informática cujos computadores possuam R e RStudio instalados

### 4.3 Inserção de Falhas na Imagem

Nesta seção de aulas iniciam-se os ensinamentos mais específicos de Processamento de Imagens aos alunos. É necessário que isso seja feita de forma lúdica e sempre com exemplos presentes em seu dia-a-dia para tornar a compreensão mais fácil. Para tal, muitas vezes não é possível utilizar os termos técnicos aprofundados, mas sim explicações mais simples e breves, com o auxílio de imagens e exemplos.

O objetivo desta seção é introduzir o conceito de imagem – além do conceito puramente intuitivo que os alunos possuem, junto com o de ruído.

Para isso, define-se textura, estrutura, níveis de cinza e outros componentes de uma imagem que forem necessários. Além disso, pode-se classificar as imagens segundo os tipos de textura. É interessante também explicar o que seriam padrões em uma imagem e como é usado hoje em dia o reconhecimento de padrões, sem defini-los tecnicamente. Por fim, define-se e classifica-se ruído em uma imagem.

Além da parte teórica, propõe-se uma aula prática com inserção de falhas em uma imagem utilizando o RStudio.

Para atingir o objetivo acima descrito, são sugeridas três aulas de 50 minutos cada uma, abordando os respectivos conteúdos:

- **Aula 7** : Elementos da Imagem
- **Aula 8** : Padrões e Ruídos da Imagem
- **Aula 9** : Inserção de Falhas na Imagem em R

### 4.3.1 Aula 7: Elementos da Imagem

Aula 7	
Professor:	
Disciplina:	
Tema:	Elementos da Imagem
Turma:	
Data:	
Duração da aula:	50 minutos

#### Objetivos

Nesta aula serão abordados aspectos relativos à imagem, como definição de estrutura, textura, níveis de cinza, classificação quanto à textura e exemplos. Sugere-se não explicar de maneira muito técnica e sim intuitiva, com auxílio de figuras para exemplificar, por exemplo, a presença de estrutura e textura na imagem. É possível abordar, levando em conta o público-alvo, aspectos da subseção 3.5.1.

Com esta aula, os alunos devem ser capazes de definir, de forma ainda intuitiva, o que é são geometria (ou estrutura) e textura de uma imagem. Também os alunos devem ser capazes de diferenciar visualmente imagens com texturas distintas e entender a complexidade de uma imagem com relação à outra, no contexto de maior quantidade de detalhes.

#### Conteúdos

- Definição de textura, estrutura, níveis de cinza e outros componentes de uma imagem que forem necessários
- Classificação das imagens quanto à textura
- Identificação de estrutura e textura em uma imagem
- Exemplos de imagens com exposição de diferentes complexidades e texturas

#### Procedimentos metodológicos

Apresentação formal do conteúdo por meio de aula expositiva com *slides*.

## Recursos didáticos

- Computador
- Projetor

### 4.3.2 Aula 8: Padrões e Ruídos da Imagem

Aula 8	
Professor:	
Disciplina:	
Tema:	Padrões e Ruídos da Imagem
Turma:	
Data:	
Duração da aula:	50 minutos

## Objetivos

O propósito desta aula, cuja teoria encontra-se na seção 3.4, é mostrar aos alunos as definições de padrões e ruídos da imagem, assim como explicar as principais fontes de ruídos, a sua classificação, exibir exemplos (como na fotografia ou na transmissão de um vídeo na televisão) e fazer comparações com ruídos de áudio. O professor também aborda, nesta aula, o que é reconhecimento de padrões e as aplicações da técnica nos dias de hoje.

É relevante falar, também, que os ruídos podem vir acidentalmente devido, por exemplo, à qualidade da máquina fotográfica, como também podem ser inseridos via *software*. Além disso, pode-se fazer uma enquete rápida se os alunos acham os ruídos em uma imagem bons ou ruins. O esperado é que a maioria diga que são ruins. É possível criar uma interação e um pequeno debate a respeito do assunto com a turma e explicar que, ao contrário do que muitos pensam, ruídos podem ser desejados (exemplo: esteganografia).

Os alunos, após esta aula, terão condições de saber o que é um padrão e um ruído no contexto de imagem. Além disso, saberão de onde vêm os ruídos, os principais tipos, a importância de, às vezes, eles serem desejados e inseridos na imagem e o que vem a ser reconhecimento de padrões em imagens.

## Conteúdos

- Definição de padrões

- Definição de ruídos e suas fontes
- Classificação dos ruídos; exemplos
- Enquete e debate sobre ruídos/falhas poderem ser desejados
- Reconhecimento de padrões e suas aplicações práticas

### Procedimentos metodológicos

Apresentação formal do conteúdo por meio de aula expositiva com *slides*.

### Recursos didáticos

- Computador
- Projetor

#### 4.3.3 Aula 9: Inserção de Falhas na Imagem em R

Aula 9	
Professor:	
Disciplina:	
Tema:	Inserção de Falhas na Imagem em R
Turma:	
Data:	
Duração da aula:	50 minutos

### Objetivos

Esta aula tem o objetivo de apresentar aos alunos maneiras de inserir falhas na imagem, em complemento ao exposto na aula anterior a respeito de fontes de ruído. Para atingir esse objetivo, o professor retorna o assunto debatido em aula anterior sobre a inserção de falhas propositadamente.

No RStudio, o professor apresenta a biblioteca **imager**, explicada no Capítulo 3, seção 3.7. Para tal, faz junto com os alunos (cada um em sua máquina ou em grupo, porém concomitantemente com o professor) a instalação da biblioteca por meio do comando **install.packages(imager)** executado no Console do RStudio.

Com a biblioteca instalada, o professor a carrega no R – no código-fonte de extensão `.r` ou na linha de comando do Console – por meio do comando `library(imager)`. Então, ele exibe algumas funções disponíveis nela e seus resultados. O código-fonte sugerido para se trabalhar em sala também foi descrito e explicado no Capítulo 3, seção 3.7.

Com esta aula, espera-se que os alunos tenham condições de instalar a biblioteca `imager` no R e programar um código em R para inserir falhas em uma imagem carregada também no R.

## Conteúdos

- Possibilidade de falhas serem desejadas e inseridas manualmente na imagem
- Instalação em conjunto com os alunos da biblioteca `imager`
- Código-fonte de inserção de falhas na imagem, conforme descrito no Capítulo 3, seção 3.7

## Procedimentos metodológicos

Aula em laboratório com programação feita pelo professor (tela do RStudio projetada) e pelos alunos, individualmente ou em grupo.

## Recursos didáticos

- Computador com R e RStudio instalados
- Projetor
- Laboratório de informática cujos computadores possuam R e RStudio instalados

## 4.4 Breve Explicação sobre Filtro Gaussiano

Esta seção contém os roteiros de aula que abordam filtro Gaussiano na imagem. Para isto, é necessário explicar o que é e para que serve um filtro na imagem, o que é a máscara usada na filtragem, quais os tipos de filtro e explicar de forma mais detalhada – sem exagerar, tendo em mente que o público-alvo é do Ensino Médio – o filtro Gaussiano.

Especificamente acerca do filtro Gaussiano, sugere-se que sejam calculadas em sala os valores de  $G(x)$  e  $G(y)$  para  $\sigma = 1$ , bem como construir com os alunos os vetores e a matriz  $G(x, y)$ . Também as aulas podem contar com breve explicação do motivo de se usar a matriz  $5 \times 5$  com o auxílio do gráfico da função Gaussiana tridimensional.

Para atingir o objetivo acima descrito, são sugeridas três aulas de 50 minutos cada uma, abordando os respectivos conteúdos:

- **Aula 10** : Filtro de Imagem
- **Aula 11** : Máscara
- **Aula 12** : Filtro Gaussiano

#### 4.4.1 Aula 10: Filtro de Imagem

<b>Aula 10</b>	
Professor:	
Disciplina:	
Tema:	Filtro de Imagem
Turma:	
Data:	
Duração da aula:	50 minutos

### Objetivos

Esta aula tem por objetivo abordar aspectos de filtro de imagem com os alunos, assim como consta na subseção 3.6.1. Com isso, sugere-se abordar o que é e para que serve o filtro, onde é usado nos dias de hoje, de preferência com figuras para exemplificar o efeito dos filtro de suavização ou daqueles usados para retirada de ruídos ou defeitos na imagem. Após isso, aborda-se a classificação dos filtros, listando os principais tipos de filtros lineares (o filtro Gaussiano deve estar na lista). Tendo em vista o público-alvo, não há necessidade de citar os filtros não lineares.

Os alunos devem ser capazes, após esta aula, de saber o que é um filtro de imagem e suas aplicações, além de identificar as diferenças entre os filtros e os impactos que eles causam na imagem.

### Conteúdos

- O que é e para que serve filtro de imagem
- Principais tipos de filtros lineares
- Usos dos filtros de imagem com exibição de figuras “antes e depois”

## Procedimentos metodológicos

Apresentação formal do conteúdo por meio de aula expositiva com *slides*.

## Recursos didáticos

- Computador
- Projetor

### 4.4.2 Aula 11: Máscara

Aula 11	
Professor:	
Disciplina:	
Tema:	Máscara
Turma:	
Data:	
Duração da aula:	50 minutos

## Objetivos

O objetivo desta aula é explicar aos alunos como os filtros funcionam. O principal componente de filtragem no contexto da imagem é a máscara. Antes de contextualizá-la, deve-se definir o que é o *pixel*.

É importante definir a máscara, esclarecer o movimento de convolução no qual a máscara passa por toda a imagem com o auxílio da figura 3.6 e explicar o que significam as entradas da matriz que representa a máscara do filtro. Vale ressaltar também que quanto maior a matriz, maior a suavização da imagem, citando exemplos simples, como o cálculo do filtro da mediana, da média ou da moda.

Vale observar em sala o que foi comentado no final na subseção 3.6.1 a respeito da reconstrução dos elementos da borda da imagem.

Com esta aula, os alunos devem ser capazes de definir, de forma ainda intuitiva, o que é *são geometria* (ou *estrutura*) e *textura* de uma imagem. Também os alunos devem ser capazes de diferenciar visualmente imagens com texturas distintas e entender a complexidade de uma imagem com relação à outra, no contexto de maior quantidade de detalhes.

## Conteúdos

- Definição de *pixel*
- Definição de máscara
- Função da máscara na filtragem
- Explicação da diferença entre uma máscara maior e uma menor
- Exemplos de cálculos dos *pixels* em filtros simples, como filtro da média, da mediana e da moda
- Observação da reconstrução da borda

## Procedimentos metodológicos

Apresentação formal do conteúdo por meio de aula expositiva com *slides*.

## Recursos didáticos

- Computador
- Projetor

### 4.4.3 Aula 12: Filtro Gaussiano

Aula 12	
Professor:	
Disciplina:	
Tema:	Filtro Gaussiano
Turma:	
Data:	
Duração da aula:	50 minutos

## Objetivos

Esta aula aborda o filtro Gaussiano. O professor pode iniciar a aula lembrando a aula anterior, na qual esse filtro foi listado nos exemplos. Após isso, explica-se como é a máscara nesse filtro, citando a média ponderada e explicando os coeficientes.



Ao explicar a origem do nome do filtro, o professor exhibe a fórmula da função Gaussiana de uma variável (não há necessidade de exhibir a fórmula da função para duas variáveis) e, então, lista as principais propriedades da função, como a simetria que leva a uma menor complexidade para o cálculo da matriz, sendo suficiente encontrar um vetor.

Também é útil o professor explicar o motivo de a matriz sempre ser quadrada e de dimensão ímpar. Após isso, o professor calcula em sala os valores das funções e dos vetores para o caso de  $\sigma = 1$ , comentando na aula porque a matriz mais usada é de ordem  $5 \times 5$ , com o auxílio do gráfico da função Gaussiana de uma ou duas variáveis.

Por fim, o professor faz a multiplicação dos vetores e encontra a matriz, cujos elementos são os elementos da máscara do filtro Gaussiano. Vale lembrar que no processo de construção dos vetores é necessário explicar o passo em que os números são multiplicados por 16 e arredondados. Sugere-se, para fins didáticos, o uso do cálculo descrito passo a passo no Capítulo 3, subseção 3.6.2.

Com esta aula, os alunos devem ser capazes de entender como funciona o filtro Gaussiano e de calcular a imagem da função Gaussiana, bem como de saber encontrar a matriz que representa a máscara do filtro, incluindo a passagem que se faz para que as entradas da matriz sejam frações de mesmo denominador, que pode ser colocado em evidência. Devido ao fato de esta aula trabalhar com matrizes, recomenda-se sua aplicação apenas nos 2º e 3º anos do Ensino Médio. Caso estejam sendo aplicadas no 1º ano, todas as aulas podem ser mantidas, exceto esta.

## Conteúdos

- Revisão de tipos de filtro
- Definição de filtro Gaussiano
- Função Gaussiana de uma variável
- Explicação da escolha e cálculo da matriz  $5 \times 5$
- Exemplos de cálculos dos *pixels* em filtros simples, como filtro da média, da mediana e da moda

## Procedimentos metodológicos

Apresentação formal do conteúdo por meio de aula expositiva com *slides*.

## Recursos didáticos

- Computador
- Projetor

## Avaliação

A avaliação, se desejada, pode ser feita pedindo-se os cálculos passo a passo para encontrar a máscara do filtro Gaussiano de dimensão 5 x 5.

### 4.5 Recuperação da Imagem

Esta última seção trata da recuperação da imagem, ou seja, retirada dos espaços inseridos na Aula 9, pelo uso da técnica do filtro Gaussiano. Recomenda-se que o professor relembre a inserção de espaços vazios na imagem com o respectivo código-fonte no qual os alunos participaram interativamente.

Então, o professor executa e explica os comandos referentes à aplicação do filtro Gaussiano na imagem com espaços com o uso da função **inpaint** do pacote **imager**, assim como descrito no Capítulo 3, seção 3.7.

Para atingir o objetivo acima descrito, é sugerida uma aula de 50 minutos.

#### 4.5.1 Aula 13: Recuperação da Imagem em R

Aula 13	
Professor:	
Disciplina:	
Tema:	Recuperação da Imagem em R
Turma:	
Data:	
Duração da aula:	50 minutos

## Objetivos

O objetivo desta aula é explicar aos alunos como se implementa em R, com o auxílio da já conhecida biblioteca **imager**, a remoção dos espaços brancos na imagem, por meio da função **inpaint** que implementa a técnica do filtro Gaussiano. Os parâmetros

de entrada são a imagem com defeito e o  $\sigma = 1$ . Sugere-se que a dinâmica seja feita com os alunos programando concomitantemente ao professor, de forma individual ou em grupo.

Com esta aula, os alunos devem ser capazes de implementar um código em R que utilize a função **inpaint** da biblioteca `imager` para retirar valores “NA” de uma imagem. Com isso, eles podem ver na prática o trabalho feito pelos filtros de suavização de imagens.

## Conteúdos

- Revisão do código-fonte da Aula 9
- Explicação da função **inpaint**
- Implementação do código-fonte de remoção de falhas conforme Capítulo 3, seção 3.7

## Procedimentos metodológicos

Aula em laboratório com programação feita pelo professor (tela do RStudio projetada) e pelos alunos, individualmente ou em grupo.

## Recursos didáticos

- Computador com R e RStudio instalados
- Projetor
- Laboratório de informática cujos computadores possuam R e RStudio instalados

# Considerações Finais

## Em relação a este trabalho

Como citado anteriormente ao longo desta dissertação, o mundo tem vivido em uma crescente necessidade de desenvolvimento nas áreas de Computação e Processamento de Imagens. Para que os jovens possuam tais habilidades, a escola exerce papel essencial e pode auxiliar por meio do ensino de técnicas básicas de Programação em uma linguagem de fácil compreensão, bem como de métodos manipulação de imagens.

Assim, este trabalho buscou, no contexto do Ensino Médio, introduzir, como atividades extraclasse, o ensino da Programação na linguagem R com a abordagem do *inpainting* – ou reconstrução de imagens – como uma das inúmeras aplicações que utilizam a programação.

Para tal, foram sugeridas 13 aulas de 50 minutos cada nas quais o professor consegue, gradualmente, ensinar de forma lúdica e, claro, superficial (tendo em vista o público-alvo). Com isso, os alunos saem do Ensino Médio para o mercado de trabalho com conhecimento básico de Programação e de Processamento Digital de Imagens, por meio da técnica de *Inpainting*.

A aplicação escolhida foi a de inserção e posterior remoção de ruídos em imagem por ser algo presente no dia-a-dia dos adolescentes e jovens atualmente. Praticamente todos possuem *smartphone* e isso torna o meio das fotografias e imagens mais próximo deles que, após a apresentação do conteúdo por meio das aulas teóricas e práticas, pode aguçar a curiosidade dos alunos, fazendo-os buscar mais informação a respeito da remoção de ruídos. Vale ressaltar que muitos já têm contato com *softwares* para tratamento de imagens.

É claro que essa não é a realidade de todos os alunos de Ensino Médio do Brasil. Junto a isso, estão as limitações para a aplicação prática das aulas sugeridas neste trabalho, já que se fazem necessários laboratório de informática para as aulas práticas, além de projetor e computador do lado do professor. Uma alternativa seria a substituição das aulas em que os alunos programam de fato por projeção em sala do computador do professor que pode fazer junto com a turma os exercícios propostos no Capítulo 4.

Vale ressaltar que, conforme explorado no Capítulo 1, as tecnologias de informação e comunicação (TICs) estão cada vez mais presentes no processo de ensino e aprendizagem, com a EaD sendo reconhecida já pelo MEC e presente em sua totalidade em algumas universidades.

## Trabalhos futuros

Tendo em vista a necessidade de modernização das técnicas de ensino e de inserção de conteúdos de Computação na grade curricular do Ensino Médio, por meio de aulas extraclasse, por enquanto, entende-se como relevante este estudo e sua continuidade. Com o passar do tempo, pode ser que a Programação seja matéria obrigatória para o Ensino Médio e, portanto, o professor pode ir além e avançar nas aplicações.

Como exemplos de possíveis continuações deste trabalho, podem-se citar:

- abordar aplicações mais complexas em R
- abordar Estatística com R
- apresentar outras linguagens de programação
- aprofundar o estudo de Processamento Digital de Imagens

O mais importante é que se busque entender a demanda dos alunos, respeitando suas limitações, pois isso varia de região para região, assim como de tempos em tempos. É um trabalho constante que o professor deve ter de avançar no que conseguir com os alunos, à procura de novos desafios que os preparem para o mercado de trabalho e para a vida, que não para e anda em constante atualização.

## Referências

- ALBALAWI, A. S. Mathematics teachers' perception of using social media in their teaching in tabuk, saudi arabia. In: . Arábia Saudita: [s.n.], 2017. v. 12. ISSN 1306-3030. Citado na página 24.
- BERTALMIO, M. et al. Image inpainting. In: . [S.l.: s.n.], 2000. p. 417–424. Citado 5 vezes nas páginas 49, 50, 55, 105 e 110.
- BERTALMIO, M. et al. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, v. 12, n. 8, p. 882–889, 2003. Citado na página 53.
- BOAS, M. L. *Mathematical methods in the physical sciences; 3rd ed.* Hoboken, NJ: Wiley, 2006. Disponível em: <<https://cds.cern.ch/record/913305>>. Citado na página 71.
- CARVALHO, A. A. de. Estudo e implementação de algoritmos clássicos para processamento digital de imagens. Lavras, Minas Gerais, 2003. Citado na página 65.
- CARVALHO, R. S. de. O uso das redes sociais no ensino de matemática através da monitoria. Paraíba, 2014. Citado na página 24.
- CHAN, T. F.; SHEN, J. Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math.*, v. 62, p. 1019–1043, 2002. Citado 4 vezes nas páginas 59, 61, 62 e 64.
- CONEXIA. *CONHEÇA OS PRÓS E CONTRAS DA TECNOLOGIA NA SALA DE AULA.* Disponível em: <<https://blog.conexiaeducacao.com.br/conheca-os-pros-e-contras-da-tecnologia-na-sala-de-aula/>>. Citado na página 31.
- CONTRERAS, R. C.; BOAVENTURA, M. Filtragem de imagens digitais e aplicações em remoção de ruídos e segmentação. *XXIV Congresso de Iniciação Científica*, São Paulo, São Paulo, 1998. Citado na página 54.
- COSTA, A. M. S. N.; FERREIRA, A. L. A. Redes sociais na educação: aprendizagem colaborativa no ensino de matemática. In: . Passo Fundo - RS: [s.n.], 2012. v. 1. Citado na página 24.
- CRIMINISI, A.; PEREZ, P.; TOYAMA, K. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, p. 1200–1212, 2004. Citado na página 53.
- DONG, J. et al. Inpainting of remote sensing sst images with deep convolutional generative adversarial network. *IEEE Geoscience and Remote Sensing Letters*, v. 16, n. 2, p. 173–177, 2019. Citado na página 109.
- ELHARROUSS, O. et al. Image inpainting: A review. *Neural Processing Letters*, Springer Science and Business Media LLC, v. 51, n. 2, p. 2007–2028, Dec 2019. ISSN 1573-773X. Disponível em: <<http://dx.doi.org/10.1007/s11063-019-10163-0>>. Citado 2 vezes nas páginas 105 e 106.

- EVANS, L. C. *Partial differential equations*. Providence, R.I.: American Mathematical Society, 2010. ISBN 9780821849743 0821849743. Citado 3 vezes nas páginas 55, 56 e 58.
- FAN, Q.; ZHANG, M. Image inpainting method based on compressed sensing. *Revista de la Facultad de Ingeniería*, v. 32, p. 412–417, 01 2017. Citado na página 111.
- FARIA, D. R. *Reconhecimento de impressões digitais com baixo custo computacional para um sistema de controle de acesso*. Dissertação (Mestrado) — Universidade Federal do Paraná, Paraná, 2005. Citado 2 vezes nas páginas 71 e 72.
- FIGUEIREDO, D. G. de. *Análise de Fourier e Equações Diferenciais Parciais*. Rio de Janeiro: IMPA, 2018. ISBN 9788524404283. Citado na página 58.
- FILHO, O. M.; NETO, H. V. *Processamento Digital de Imagens*. Rio de Janeiro: Brasport, 1999. ISBN 8574520098. Citado na página 54.
- GARRISON, D. Three generations of technological innovations in distance education. *Distance Education - DISTANCE EDUC*, v. 6, p. 235–241, 09 1985. Citado na página 18.
- GILBARG, D.; TRUDINGER, N. S. *Elliptic Partial Differential Equations of Second Order*. [S.l.]: Springer, 2001. ISBN 978-3-642-96379-7. Citado na página 62.
- GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008. ISBN 9780131687288 013168728X 9780135052679 013505267X. Citado 3 vezes nas páginas 64, 66 e 68.
- HAN, X. et al. Finet: Compatible and diverse fashion image inpainting. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2019. p. 4480–4490. Citado na página 109.
- HIRANI, A. N.; TOTSUKA, T. Combining frequency and spatial domain information for fast interactive image noise removal. In: . [S.l.: s.n.], 1996. p. 269–276. Citado na página 52.
- JUNIOR, R. I.; IORIO, V. de M. *Equações Diferenciais Parciais*. Rio de Janeiro: IMPA, 2013. ISBN 9788524400353. Citado na página 58.
- KLEIN, T. E.; GALLAGER, R. G. Power control for the additive white gaussian noise channel under channel estimation errors. In: *Proceedings. 2001 IEEE International Symposium on Information Theory (IEEE Cat. No.01CH37252)*. [S.l.: s.n.], 2001. p. 304–. Citado na página 54.
- LI, J.; SONG, G.; ZHANG, M. Occluded offline handwritten chinese character recognition using deep convolutional generative adversarial network and improved googlenet. *Neural Computing and Applications*, v. 32, 11 2018. Citado na página 109.
- LI, Q.; HAN, Y.; DANG, J. Image decomposing for inpainting using compressed sensing in dct domain. *Frontiers of Computer Science*, v. 8, 12 2014. Citado 3 vezes nas páginas 109, 110 e 111.

- LIU, H. et al. Image inpainting based on generative adversarial networks. In: *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. [S.l.: s.n.], 2018. p. 373–378. Citado na página 109.
- MACEDO, D. F. et al. A importância da utilização do aplicativo geogebra em aulas de matemática: experiência vivenciada em uma escola da educação básica. *Anais do IV Congresso Nacional de Educação*, João Pessoa, Paraíba, 11 2017. Citado na página 27.
- MCKIMM, J.; JOLLIE, C.; CANTILLON, P. Web based learning. *BMJ*, BMJ Publishing Group Ltd, v. 326, n. 7394, p. 870–873, 2003. ISSN 0959-8138. Disponível em: <<https://www.bmj.com/content/326/7394/870>>. Citado na página 16.
- MILLIGAN, C. Delivering staff and professional development using t using virtual learning environments. in: ts. the role of he role of virtual learning environments in the online delivery of ts in the online delivery of staff development. 10 1999. Disponível em: <<http://www.icbl.hw.ac.uk/jtap-573/573r2-3.html>>. Citado na página 16.
- NG, R.; LATIF, L. A. Social media and the teaching of mathematics in a lifelong learning environment. In: . Malaysia: [s.n.], 2011. Citado na página 24.
- NIPPER, S. Third generation distance learning and computer conferencing. In: *Robin Mason Anthony Kaye (eds.), MINDWAVE: Communication, Computers and Distance Education*. [S.l.: s.n.], 1989. p. 63–73. Citado na página 18.
- NOVELLO, T. *Investigando a interação das professoras no Ambiente Virtual Mathemolhes*. Dissertação (Mestrado) — Universidade Federal do Rio Grande, Rio Grande, 2006. Citado na página 16.
- PEDRINI, H.; SCHWARTZ, W. R. *Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações*. São Paulo: Thomson, 2007. ISBN 9788522105953. Citado na página 69.
- PEREIRA, T. de L. M. *O Uso do Software Geogebra em uma Escola Pública: interações entre alunos e professor em atividades e tarefas de geometria para o ensino fundamental e médio*. Dissertação (Mestrado) — Universidade Federal de Juiz de Fora, Juiz de Fora, Minas Gerais, 2012. Citado na página 27.
- RIBEIRO, B. *Suavização de Imagens*. Rio de Janeiro, 2006. Citado na página 54.
- SAHA, R. A.; AYUB, A. F. M.; TARMIZI, R. A. The effects of geogebra on mathematics achievement: Enlightening coordinate geometry learning. *Procedia - Social and Behavioral Sciences*, v. 8, p. 686 – 693, 2010. ISSN 1877-0428. International Conference on Mathematics Education Research 2010 (ICMER 2010). Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877042810022007>>. Citado na página 26.
- SANCHES, C. H. et al. Técnicas de suavização de imagens e eliminação de ruídos. *Anais do EATI*, v. 1, n. 5, p. 21–30, 11 2015. Citado 2 vezes nas páginas 71 e 72.
- SCHONLIEB, C.-B. *Partial Differential Equation Methods for Image Inpainting*. [S.l.]: Cambridge University Press, 2015. (Cambridge Monographs on Applied and Computational Mathematics). Citado 2 vezes nas páginas 58 e 59.



- SCHULZ, E. A. B. d. S. A.; VELHO, L. Compressive sensing. In: . Rio de Janeiro: [s.n.], 2009. v. 6. Citado na página 110.
- SEARA, D. M. Visão geral de detecção de bordas. Santa Catarina, Paraná, 1998. Citado na página 54.
- SHIN, Y.-G. et al. Pepsi++: Fast and lightweight network for image inpainting. *IEEE Transactions on Neural Networks and Learning Systems*, Institute of Electrical and Electronics Engineers (IEEE), p. 1–14, 2020. ISSN 2162-2388. Disponível em: <<http://dx.doi.org/10.1109/TNNLS.2020.2978501>>. Citado na página 109.
- SILVA, C. dos S.; SANTIBANEZ, M. R. F.; ZUASNABAR, D. M. H. Geometria analítica com geogebra usando a teoria dos campos conceituais de vergnaud no ensino médio. *Anais do SENID*, Paraná, v. 1, n. 5, 2018. ISSN 2238-5916. Citado na página 26.
- SILVA, G. M. da. Um estudo sobre o uso do geogebra na aprendizagem de geometria analítica no ensino médio. São Carlos, São Paulo, 2016. Citado na página 26.
- SIQUEIRA, D. N. de; CAETANO, J. J. O uso do geogebra no ensino de funções no ensino médio. *Os desafios da escola pública paranaense na perspectiva do professor PDE*, Paraná, v. 1, 2016. Citado 2 vezes nas páginas 26 e 27.
- STOLOJESCU-CRISAN, C.; ISAR, A. Images compressive sensing reconstruction by inpainting. In: . [S.l.: s.n.], 2015. p. 1–4. Citado 2 vezes nas páginas 109 e 110.

# APÊNDICE A – Técnicas de Recuperação de Imagens

Este apêndice tem por objetivo aprofundar um pouco algumas técnicas utilizadas no processo de recuperação de imagens.

## A.1 Outras Abordagens de *Inpainting*

De acordo com BERTALMIO et al., 2000, os algoritmos de *inpainting* existentes dividem-se em três grandes grupos, relacionados a:

1. restauração de filmes – incluindo remoção de ruído
2. síntese de textura
3. desobstrução

A maioria dos trabalhos em restauração de imagens concentra-se no primeiro grupo, que leva o nome mais intuitivo e inclui a remoção de alguma deterioração na imagem, como texto, ruído ou mancha. A remoção de ruído é a principal aplicação da técnica e, por isso, é o que mais se encontra na literatura e nos *softwares*.

A última categoria não é muito explorada mas tem sido cada vez mais utilizada e estudada. Consiste em retirar um objeto da cena original, reconstruindo a imagem ao fundo. A principal razão de esta categoria não ser tão popular é a dificuldade de se preencher falhas maiores de uma imagem com formulações tradicionais, tendo em vista que seu escopo é limitado a regiões onde são conhecidas as vizinhanças. Nos problemas de desobstrução, no entanto, nem sempre é possível ter tais informações.

Interpolações bilineares e bicúbicas podem ser usadas para interpolar os espaços de objetos a retirar. Também são possíveis interpolações mais avançadas. Como em outros objetivos de retoque digital, são utilizados filtros. A filtragem de passagem única tem um desfoque indesejado das regiões pintadas, mas a maioria dos esquemas de pintura aplica os filtros repetidas vezes, bem como combina diversos tipos distintos.

De acordo com ELHARROUSS et al., 2019, os métodos de *inpainting* são abordados de três principais formas:

1. Baseadas em sequências

2. Baseadas em CNN (*Convolutional Neural Networks*)
3. Baseadas em GAN (*Generative Adversarial Network*)

A abordagem baseada em sequências é a mais tradicional e foi a utilizada nos primeiros trabalhos da área. Nela, é possível dividir os algoritmos de *inpainting* em dois grupos: os baseados em *patch* e os baseados em difusão.

Os algoritmos baseados em *patch* – ou, neste contexto, fragmentos – são técnicas para preencher a região perdida, remendo por remendo, procurando por fragmentos de substituição (ou seja, fragmentos candidatos) na parte não danificada que correspondam bem à porção a ser retocada para, então, copiá-los para os locais correspondentes. Muitos métodos foram propostos para a reconstrução de imagens usando métodos baseados em *patch*.

O termo **difusão** surgiu da ideia de propagar localmente a informação por analogia com fenômenos físicos, como a propagação de calor e de ondas, via equações diferenciais parciais. A redução de ruído foi uma das primeiras aplicações deste conceito na área de Processamento de Imagem. Métodos baseados em difusão preenchem a região ausente propagando suavemente o conteúdo da imagem desde as fronteiras até o interior da região faltante.

Abaixo segue uma tabela, na Figura A.1, levantada por ELHARROUSS et al., 2019 que lista diversos trabalhos em que o método segue a abordagem baseada em sequências.

Table 1: Sequential-based method for image inpainting.

Category	Method	Feature	Image
Patch-based	Muddala et al. 2016 [1]	layered depth image (LDI)	RGB
	ISOGAWA et. 2018 [2]	Super-pixel	RGB
	Ružić et al. 2014 [3], Liu et al. 2018 [9]	Markov Random Field (MRF)	RGB
	Jin et al. 2015 [4]	Annihilation property filter, low rank structured matrix	RGB
	Kawai et al. 2015 [5]	Background geometry estimation	Artificial
	Guo et al. 2018 [6]	Two-Stage Low Rank Approximation (TSLRA)	RGB
	Xue et al. 2017 [7]	Low Gradient Regularization	Depth
	Ding et al. 2019 [11]	Nonlocal Texture Matching, Nonlinear Filtering ( $\alpha$ -trimmed mean filter)	RGB
	Duan et al. 2015 [12]	non-local Mumford–Shah model (NL-MS)	RGB
	Fan et al. 2018 [13]	Sum of Squared Differences (SSD)	Gray scale
	Jiang et al. 2016 [14]	Canny filter, Segmentation	RGB
	Alilou et al. 2017 [15]	Singular value decomposition and an approximation matrix	RGB
	Lu et al. 2018 [17]	Gradient-based low rank approximation	Gray scale
	Wang et al. 2017 [20], Yao et al. 2018 [22]	Structure and texture analysis	Thangka images
	Wei et al. 2016 [21]	Structure-aware	RGB
Zeng et al. 2019 [23]	Saliency Map and Gray entropy	RGB	
Zhang et al. 2018 [24]	joint probability density matrix (JPDM).	RGB	
Diffusion-based	Li et al. 2017 [8]	Intra-channel and Inter-channel local variances	RGB
	Li et al. 2016 [16]	Distance and direction between the damaged pixel and its neighborhood pixels.	RGB
	Sridevi et al. 2019 [19]	Fractional-order derivative and Fourier transform	Gray scale

Fonte: ELHARROUSS et al., 2019

Figura A.1 – Tabela com métodos baseados em sequências

Esses métodos são eficientes para imagens simples e suaves. Já para imagens complexas, com muita textura ou com um objeto a ser retirado que cubra uma parte não homogênea, pode ser difícil procurar um pedaço semelhante que sirva para substituir ou propagar o que falta (seja via *patch* ou difusão).

Com o avanço da pesquisa em redes neurais, surgiram as CNN (*Convolutional Neural Networks*), as chamadas Redes Neurais Convolucionais, do tipo *feed-forward*, que vêm sendo aplicadas com sucesso na área de *inpainting*.

As CNNs são usadas especificamente para melhorar os resultados esperados neste campo, usando dados de treinamento em larga escala. Os métodos baseados em sequências são usados em algumas partes da reconstrução da imagem, como no preenchimento de detalhes de textura, mas ainda permanece o problema de capturar a estrutura global. Vários métodos foram propostos para a reconstrução de imagens usando redes neurais convolucionais. Com isso, os métodos recuperam o bloco ausente com boa precisão em termos de estrutura e textura.

Table 2: CNN-based method for image inpainting.

Method	Architecture	Data Type
Yan et al. 2018 [25], Chang et al. 2019 [28], Zhu et al 2018 [30], Liu et al. 2019 [33], Hsu et al. 2017 [36], Liao et al. 2019 [40], Hertz et al. 2019 [42]	Encoder-decoder network	RGB
Weerasekera et al 2018 [26]	CNN	TGB-D
Zhao et al. 2018 [27]	CNN	Y-ray
Cai et al. 2017 [29]	Blind CNN	Grayscale
Sidorov et al. 2019 [31]	3D CNN	RGB
Zeng et al. 2019 [32]	Pyramid-context encoder network	RGB
Pathak et al. 2016 [34]	Context-encoder, CNN	RGB
Sasaki et al 2017 [35]	CNN	line drawing images
Nakamura et al. 2017 [37]	CNN	Image with text
Xiang et al. 2017 [39]	CNN, GAN	Damaged old pictures
Cai et al. 2018 [41]	CNN	RGB

Fonte: ELHARROUSS et al., 2019

Figura A.2 – Tabela com métodos baseados em redes neurais convolucionais

As Redes Adversárias Generativas, ou GANs, do inglês *Generative Adversarial Network*, são arquiteturas profundas com duas redes neurais do tipo *feed-forward*: uma representa o gerador e a outra é o discriminador.

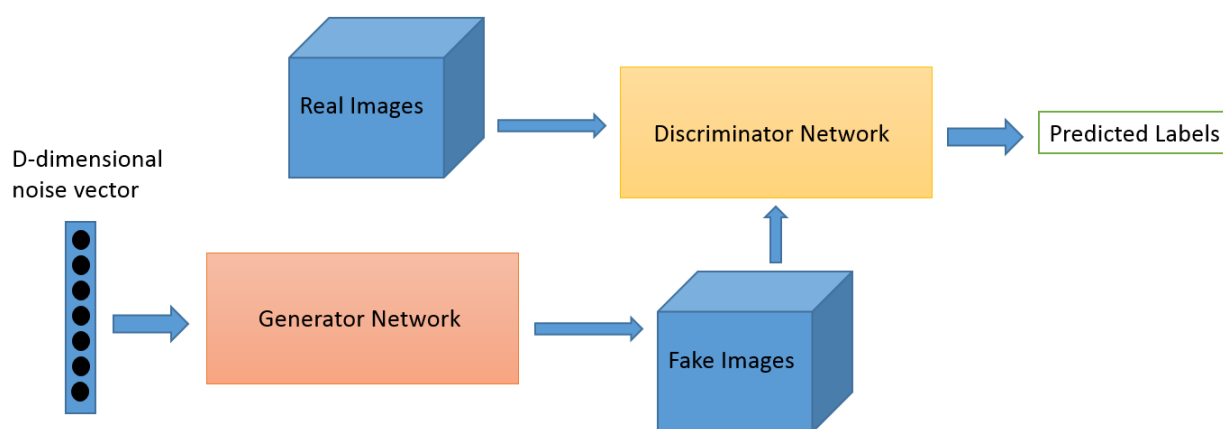
O gerador recebe ruídos aleatórios como entrada e gera algumas amostras falsas semelhantes às reais; já o discriminador é treinado a determinar se as amostras são reais ou falsas. Atualmente, as GANs são a técnica mais usada em todos os aplicativos de visão

computacional.

Como é a técnica mais rebuscada e recente, poucos são os trabalhos que a utilizam no processo de retoque de imagem digital. Alguns desses trabalhos estão listados a seguir e suas referências encontram-se, respectivamente, em LI; SONG; ZHANG, 2018, SHIN et al., 2020, DONG et al., 2019, HAN et al., 2019 e LIU et al., 2018.

- proposta de *inpainting* e reconhecimento de caracteres chineses oclusos
- projeto PEPSI++
- método para reconstruir imagens do fundo do mar
- abordagem para completar imagens de moda, conhecida por FiNet (*Fashion Inpainting Networks*)
- método para retocar imagens faciais

A Figura A.3 exibe um esquema visual de como é feito o processo utilizando gerador e discriminador com redes adversárias generativas.



Fonte: <http://deeplearningbook.com.br/>

Figura A.3 – Esquema de funcionamento das redes GANs

## A.2 Compressive Sensing

Outra técnica moderna e altamente relevante na área de recuperação de imagens é o *Compressive Sensing* ou *Compressed Sensing/Sampling*, que surgiu da necessidade de se comprimir imagens de forma eficiente, ou seja, sem perder informações relevantes. Conforme STOLOJESCU-CRISAN; ISAR, 2015 e LI; HAN; DANG, 2014, o objetivo da técnica é reduzir a frequência de amostragem na digitalização de um sinal analógico.

Vale lembrar que amostrar um sinal é o processo no qual se converte um sinal – representado, normalmente, por uma função contínua no tempo ou no espaço – em uma sequência numérica (uma função discreta). Então, para se ter uma representação discreta de um sinal contínuo no tempo, realiza-se amostragem periódica deste sinal.

O Teorema da amostragem de Nyquist diz que, apesar de o processo de amostragem sempre gerar perdas com relação ao sinal original, o espectro do sinal não será afetado pela sua discretização se esta ocorrer com uma frequência de amostragem de, pelo menos, duas vezes a maior frequência do sinal. Ao dobro da máxima frequência do sinal dá-se o nome de limite de Nyquist.

Os algoritmos de *Compressive Sensing* são robustos e permitem amostragens a taxas muito menores que o limite de Nyquist e recuperação dos sinais com pouca perda por meio de otimização convexa com a norma  $\ell_1$  a ser minimizada sujeita a algumas restrições. Ressalta-se que, dado um vetor  $\mathbf{v} = (v_1, \dots, v_m) \in \mathbb{C}^m$ , a norma  $\ell_1$  é definida como

$$\|\mathbf{v}\|_{\ell_1} = \sum_{k=1}^m |v_k|.$$

O trabalho de SCHULZ; VELHO, 2009 cita duas abordagens para o *Compressive Sensing* (CS), sendo a segunda mais aplicável à maioria dos sinais analógicos por não serem esparsos, ou seja, não possuírem função densidade de probabilidade como uma função Laplaciana:

- CS Básico: teoria que estipula restrições para a recuperação exata de sinais esparsos nos quais as medidas não estão corrompidas por ruído
- CS Robusto: expansão da abordagem anterior para possibilitar aplicações de CS a sinais que não são exatamente esparsos e cujas medidas estão corrompidas por ruído

O problema de reconstrução por CS baseados na norma  $\ell_1$  pode ser visto como um problema de preenchimento de *pixels* faltantes nas imagens obtidas após o processo. Algoritmos de *inpainting* são usados para esse objetivo, logo, podem ser vistos como algoritmos de reconstrução por CS.

Em STOJESCU-CRISAN; ISAR, 2015 é proposto um algoritmo de reconstrução de imagens baseado em Equações Diferenciais Parciais Elípticas para a interpolação das imagens obtidas no processo de CS.

Já LI; HAN; DANG, 2014 propõe um algoritmo de *inpainting* (estrutural e textural) que utiliza CS no domínio da frequência para reconstruir imagens corrompidas. Tal como BERTALMIO et al., 2000, a imagem é decomposta em duas partes: estrutural e textural. Conforme concluído por LI; HAN; DANG, 2014, a implementação com CS

apresentou vantagens quando comparada às técnicas clássicas de *inpainting* em imagens com muita informação perdida, sem um padrão muito parecido da porção danificada no resto da imagem.

No entanto, o método proposto por LI; HAN; DANG, 2014 tem algumas desvantagens. Primeiramente, a reconstrução com o uso de CS é feita em toda a imagem (parte danificada e parte não danificada), fazendo com que a parte “boa” da imagem não fique com a qualidade que fica nos métodos convencionais, já que estes mantêm a parte não danificada intacta. Em segundo lugar, quando a parte faltante fica concentrada em uma parte sem muito gradiente de textura, o método proposto tem mais custo computacional que uma simples interpolação e não traz benefícios de *performance*.

Outro algoritmo de reconstrução de imagens baseado em CS foi proposto por FAN; ZHANG, 2017, também no domínio da frequência, e apresentou bons resultados.

O *Compressive Sensing* vem surgindo cada vez mais em pesquisas na área de reconstrução de imagens como uma nova técnica eficiente e menos custosa que ainda possui muito a ser explorado para ampliar os horizontes no que diz respeito ao *Inpainting*.