

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS DE DOIS VIZINHOS
CURSO DE ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS

KÊNIA SAMARA MOURÃO SANTOS

**CONSTRUÇÃO DE OBJETOS ESPACIAIS FUZZY A PARTIR DE
IMAGENS DE SATÉLITE**

TRABALHO DE CONCLUSÃO DE CURSO DE ESPECIALIZAÇÃO

DOIS VIZINHOS
2022

KÊNIA SAMARA MOURÃO SANTOS

CONSTRUÇÃO DE OBJETOS ESPACIAIS FUZZY A PARTIR DE IMAGENS DE SATÉLITE

Trabalho de Conclusão de Curso de Especialização apresentado ao Curso de Especialização em Ciência de Dados da Universidade Tecnológica Federal do Paraná, como requisito para a obtenção do título de Especialista em Ciência de Dados.

Orientador: Prof. Dr. Anderson Chaves Carniel

DOIS VIZINHOS
2022



4.0 Internacional

Esta licença permite remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

KÊNIA SAMARA MOURÃO SANTOS

**CONSTRUÇÃO DE OBJETOS ESPACIAIS FUZZY A PARTIR DE
IMAGENS DE SATÉLITE**

Trabalho de Conclusão de Curso de Especialização apresentado ao Curso de Especialização em Ciência de Dados da Universidade Tecnológica Federal do Paraná, como requisito para a obtenção do título de Especialista em Ciência de Dados.

Data de aprovação: 09/junho/2022

Anderson Chaves Carniel
Doutorado
Universidade Federal de São Carlos

Rafael Alves Paes de Oliveira
Doutorado
Universidade Tecnológica Federal do Paraná - Câmpus Dois Vizinhos

Yuri Kaszubowski Lopes
Doutorado
Universidade do Estado de Santa Catarina

DOIS VIZINHOS
2022

AGRADECIMENTOS

Agradeço o professor Anderson Carniel pela orientação e paciência ao longo do desenvolvimento deste trabalho. Agradeço também a empresa Agrosatélite por custear parte do curso de especialização em Ciência de Dados.

RESUMO

Cada vez mais a lógica e a teoria de conjuntos *fuzzy* vêm sendo empregadas de forma satisfatória em ambientes de Sistemas de Informações Geográficas (SIG). A representação de tipos de dados espaciais usando a teoria de conjuntos *fuzzy* tem levado a criação e definição dos tipos de dados espaciais *fuzzy*. Algumas implementações que possibilitam o uso de objetos espaciais *fuzzy* vêm sendo introduzida, como o pacote *fsr*, objetivando permitir o uso de Álgebras Espaciais *Fuzzy*, como a Álgebra Espacial *Plateau* (SPA). Nesse contexto, existe já um caminho percorrido e consolidado para a construção de objetos espaciais *fuzzy* a partir de conjuntos pontos. No entanto, ainda há o desafio de construir tais tipos de objetos a partir de dados espaciais matriciais, com destaque para imagens de satélite. Visando preencher essa lacuna, o presente trabalho apresenta a inclusão de uma política de fuzzificação que permite construir objetos espaciais *fuzzy* baseados na SPA a partir de dados matriciais. Para isso, foi utilizado um tipo de *soft classification* chamado Modelo Linear de Mistura Espectral, na qual atribui graus de pertinência de cada pixel para cada uma das classes de mapeamento contida em uma imagem. O resultado foi incorporado ao pacote *fsr* na função *spa_creator* para a geração de objetos espaciais *fuzzy* e possibilitar seu uso em ambientes SIG.

Palavras-chave: Modelo Linear de Mistura Espectral; Política de Fuzzificação; dados espaciais *Plateau*; Imagem de satélite; SIG.

ABSTRACT

Fuzzy logic and theory have received increasing interest in Geographic Information Systems (GIS) environments. The representation of spatial data types using fuzzy set theory has led to the creation and definition of spatial fuzzy data types. Some implementations that enable the use of fuzzy spatial objects have been introduced, such as the *fsr* package, aiming to enable the use of fuzzy spatial algebras, such as the Spatial Plateau Algebra (SPA). In this context, there is already a path taken and consolidated for the construction of fuzzy spatial objects from point sets. However, there is still the challenge of building such types of objects from matrix spatial data, especially satellite images. Aiming to fill this gap, this work presents the inclusion of a fuzzification policy that allows to build SPA-based fuzzy spatial objects from matrix data. For this, a type of soft classification called Linear Spectral Mixture Model was used, which assigns membership degrees of each pixel to each of the mapping classes contained in an image. The result was incorporated into the *fsr* package in the *spa* creator function to generate fuzzy spatial objects and enable their use in GIS environments

Keywords: Linear Spectral Mixture Model; Fuzzy policy; Spatial plateau data; Satellite image; GIS.

SUMÁRIO

1	INTRODUÇÃO	7
1.1	Contexto	7
1.2	Justificativa	8
1.3	Objetivos	8
1.4	Contribuições	8
1.5	Organização do Trabalho	9
2	FUNDAMENTAÇÃO TEÓRICA	10
2.1	Dados Espaciais	10
2.1.1	Dados vetoriais	10
2.1.2	Dados matriciais	11
2.2	Dados espaciais fuzzy	11
2.3	Construção de objetos espaciais fuzzy	15
2.4	Modelo de Mistura Espectral	15
3	CONSTRUÇÃO DE OBJETOS ESPACIAIS FUZZY A PARTIR DE DADOS MATRICIAIS	18
3.1	Visão Geral	18
3.2	Inclusão de uma nova política de fuzzificação	19
3.3	Incorporação da política no pacote fsr	20
4	APLICANDO A NOVA POLÍTICA DE FUZZIFICAÇÃO EM UMA SITUAÇÃO REAL	24
4.1	Contexto da aplicação e obtenção das imagens de satélite	24
4.1.1	Obtenção da imagem de satélite	25
4.1.2	Obtenção dos <i>endmembers</i>	26
4.2	Aplicação do método de fuzzificação (aplicação da função <i>mesma_policy</i>) e política de construção	26
5	CONCLUSÕES E TRABALHOS FUTUROS	30
	REFERÊNCIAS	31

1 INTRODUÇÃO

1.1 Contexto

Os Sistemas de Informação Geográfica (SIG) são bastante utilizados na representação espacial e seu potencial é gigantesco quando vinculado ao uso de bancos de dados espaciais (CARNIEL; SCHNEIDER, 2019). Os tipos de representação espacial utilizados nos SIGs são linhas, polígonos e pontos, que possuem características bem definidas, como bordas sólidas (*crisp*). Alguns exemplos de utilização inclui a representação dos limites de fazendas, municípios e estados como polígonos (ou regiões).

Entretanto, alguns tipos de dados espaciais não podem ser representados desta forma, pois não possuem uma exatidão ou contorno muito bem delimitados. Isso pode ser devido a diversos fatores, como aquisição de dados imprecisos, representação incompleta, mudanças dinâmicas e imprecisões inerentes aos fenômenos espaciais (LIU et al., 2015). Em espaços contínuos onde é difícil definir claramente as bordas dos objetos, termos como 'próximo de' e outros termos que demonstram incertezas, podem expressar melhor algumas divisões do que a simples classificação binária (JASIEWICZ, 2011). Essas imprecisões espaciais são um tipo de imperfeição que surge na presença de casos limítrofes, locais que não pertencem inteiramente nem a uma classe e nem à outra, e são chamados de *objetos espaciais vagos*. Esses objetos não podem ser representados por dados espaciais *crisp*, pois a sua característica de vagueza espacial não pode ser negligenciada (CARNIEL, 2014). Pode-se mencionar como exemplo o caso do mapeamento de um tipo de vegetação. Pode ser difícil decidir no mapeamento a qual classe um determinado local pertence, se a um tipo de vegetação ou a outro, uma vez que a transição entre classes pode ocorrer de forma gradual. Esse tipo de situação pode acontecer em diversos casos reais. Muitas teorias têm sido propostas para resolver casos de imprecisões. A teoria *fuzzy* é a que mais vem sendo utilizada em diversos campos SIGs, como classificação de imagem, mapeamento do solo, turismo, etc. (JASIEWICZ, 2011). Dessa forma, *tipos de dados espaciais fuzzy* e operações espaciais *fuzzy* vem sendo definidos na literatura (DILO; BY; STEIN, 2007; CARNIEL; SCHNEIDER, 2019).

Existe um esforço para utilização de objetos espaciais *fuzzy* em ambientes SIG ou bancos de dados espaciais. Um desses esforços é o pacote *fsr* (CARNIEL et al., 2021), no qual, por meio da função `spa_creator` possibilita a criação de objetos espaciais *fuzzy* representados conforme a *Spatial Plateau Algebra* (SPA) (CARNIEL; SCHNEIDER, 2018). Uma vez criados, tais objetos podem ser manipulados em ambientes de bancos de dados espaciais e aplicações de ciência de dados espaciais. A construção de tais objetos foi formulada considerando que a origem de dados seja um conjunto de pontos, de certa forma negligenciando outra fonte importante de dados espaciais: o formato matricial (imagens de satélite, por exemplo). Diante disso, o objetivo deste trabalho consiste na criação de uma nova abordagem de construção de

objetos espaciais *fuzzy* a partir de imagens de satélite.

1.2 Justificativa

O uso de lógica *fuzzy* vem se tornando cada vez mais difundida no ambiente de SIG (DILO; BY; STEIN, 2007; CARNIEL, 2014). Nesse contexto, existem diversas políticas de fuzzificação utilizadas para construção de objetos espaciais *fuzzy* (CARNIEL; SCHNEIDER, 2019), como as implementadas pelo pacote *fsr* (CARNIEL et al., 2021). Entretanto, as políticas existentes são voltadas para objetos vetoriais, desconsiderando a existência de dados em formato matricial. Imagens de satélite, por exemplo, são excelentes fontes de dados espaciais, uma vez que são compostas por pixels com informações espectrais e espaciais importantes para o entendimento de um território. Uma das aplicações desse tipo de dado é a obtenção de mapas por meio de *classificação dos pixels*. As classificações consistem na atribuição de valores de pertencimento ou não pertencimento de um determinado pixel à uma classe. No entanto, esse tipo de classificação não leva em consideração zonas de transição de classes, pois consistem em classificações do tipo *crisp*.

Nesse sentido, o uso de classificadores em imagens de satélite com características *fuzzy*, ou seja, com características de *soft classification* como políticas de fuzzificação possibilitam obter dados mais detalhados sobre graus de pertinência dos pixels às classes de interesse. Os Modelos Lineares de Mistura Espectral (SHIMABUKURO; PONZONI, 2017), por exemplo, são um dos tipos de classificação suave, ou *soft classification*, (FERREIRA, 2003) voltadas à imagens de satélite que atribuem valores de 0 a 1 a cada pixel da imagem de acordo com a presença de uma determinada classe nas respostas espectrais das bandas que formam a imagem. Diante disso, construir objetos espaciais *fuzzy* a partir desse tipo de classificação pode resultar em níveis de detalhamento importantes para análises espaciais.

1.3 Objetivos

O objetivo geral deste Trabalho de Conclusão de Curso (TCC) é desenvolver uma nova abordagem de construção de objetos espaciais *fuzzy* baseados na SPA a partir de imagens de satélite por meio do uso do Modelo Linear de Mistura Espectral. Como objetivos específicos, este TCC visa:

- Implementar um método que permita a construção de objetos espaciais *fuzzy* baseados na SPA a partir de imagens de satélite;
- Incorporar este método no pacote *fsr*;
- Formular um estudo de caso para mostrar a aplicabilidade do método implementado;

1.4 Contribuições

As contribuições deste trabalho são listadas a seguir:

- Desenvolvimento de um método baseado no Modelo Linear de Mistura Espectral para a construção de objetos espaciais *fuzzy* a partir de imagens de satélite;
- Integração do método implementado no pacote *fsr*; e,
- Condução de um estudo de caso baseado em uma imagem real de satélite para mostrar a utilização prática do método incorporado no pacote *fsr*.

1.5 Organização do Trabalho

O trabalho está dividido da seguinte forma: Capítulo 1 consiste na introdução e apresentação dos objetivos do trabalho. No Capítulo 2 está apresentada a fundamentação teórica necessária para a compreensão deste TCC. No Capítulo 3 é discutido o método de construção de objetos espaciais *fuzzy* e as implementações adicionadas. No Capítulo 4 ilustra um exemplo de aplicação das implementações realizadas em uma situação real. Por fim, o Capítulo 5 conclui este TCC.

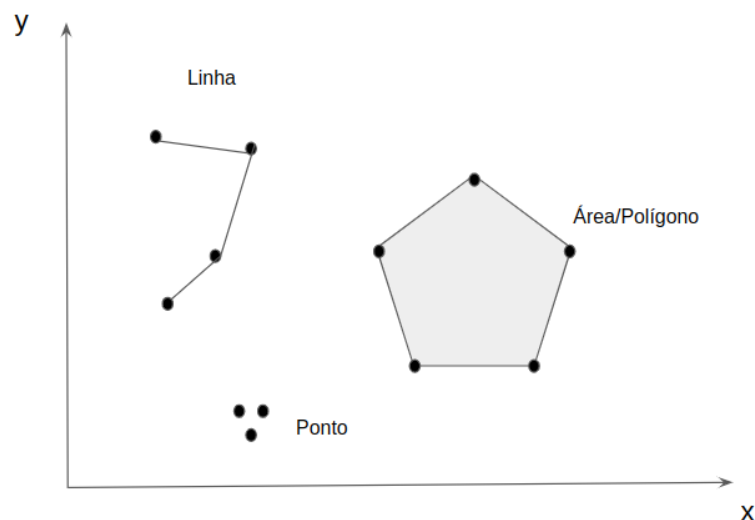
2 FUNDAMENTAÇÃO TEÓRICA

2.1 Dados Espaciais

2.1.1 Dados vetoriais

No ambiente de Sistema de Informação Geográfica (SIG) é possível fazer uma representação de espaços contínuos do mundo real através de dados vetoriais. Essa representação é feita por meio da utilização de objetos espaciais (RENZHONG, 1998), que comumente usam dados espaciais *crisp* no espaço Euclidiano bidimensional ou tridimensional (SCHNEIDER; BEHR,). As estruturas vetoriais *crisp* utilizadas para representar as coordenadas das fronteiras de cada entidade geográfica são: pontos, linhas e polígonos (CÂMARA, 2005), que buscam mostrar os limites espaciais de forma nítida. Tais objetos são caracterizados por sua localização, fronteira, interior e extensão precisamente definidos no espaço. A Figura 1 ilustra os tipos de objetos espaciais *crisp*.

Figura 1 – Exemplos de objetos espaciais *crisp*.



Fonte: Adaptado de Câmara (2005)

Câmara (2005) descreve *ponto* como sendo um par ordenado (x,y) de coordenadas espaciais, e pode ser utilizado para identificar localizações ou ocorrências no espaço, como localização de espécies vegetais, crimes e doenças. A *linha*, por sua vez, é um conjunto de pontos conectados, e são utilizadas para armazenar feições unidimensionais, como uma estrada ou um rio. A *área* (*região*), ou *polígono*, é a região do plano limitada por uma ou mais linhas poligonais conectadas, de tal forma que o último ponto da linha seja idêntico ao primeiro da próxima, de forma a estabelecer uma fronteira entre duas regiões: a interior e a exterior do

polígono. Os polígonos são utilizados para representação de unidades espaciais individuais, como estados, países, distritos e zonas.

Apesar dos objetos espaciais *crisp* serem consolidados em ambientes SIG, eles não representam as imprecisões espaciais, ou seja, a realidade espacial em alguns casos, como precisão inexatas, limites vagos e imprecisos (CARNIEL; SCHNEIDER, 2019), como a transição de classes de vegetação, tipos de solo, poluição, etc. (DILO; BY; STEIN, 2007). Com finalidade de resolver esse tipo de problema, muitos trabalhos vêm desenvolvendo métodos baseados em lógica *fuzzy*, usado para modelar esses objetos espaciais difusos (CARNIEL; SCHNEIDER, 2019).

2.1.2 Dados matriciais

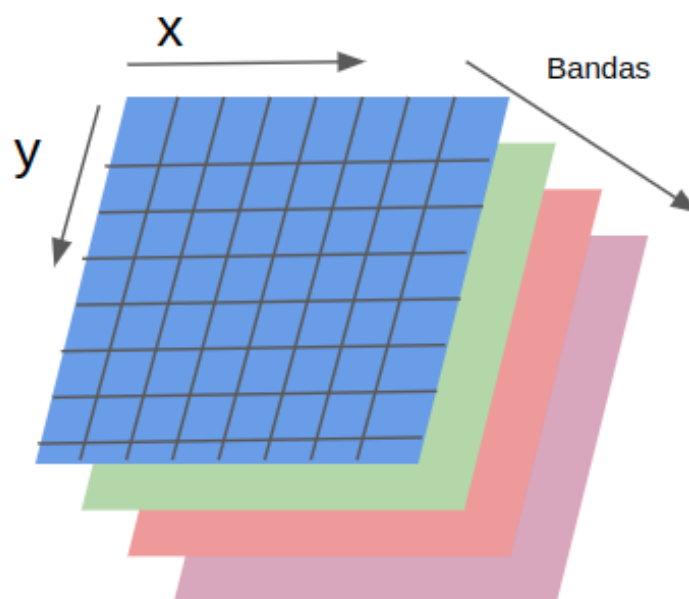
No sensoriamento remoto, os sensores espectrais registram a energia refletida ou emitida de um objeto ou área de interesse em múltiplas bandas espectrais (canais, regiões) do espectro eletromagnético. Boa parte desses sensores coleta tais dados em formato matricial (JENSEN; EPIPHANIO, 2009). Os dados matriciais podem ser representados como uma matriz com m linhas e n colunas. Cada posição da matriz é equivalente a um *pixel*. Cada pixel possui uma localização dentro da matriz (coordenadas no eixo x e y ou coordenadas geográficas em casos de dados matriciais cartográficos) e um ou mais valores de brilho associado. O conjunto de dados pode ser composto por k bandas individuais (associado à quantidade de regiões do espectro eletromagnético que o sensor é capaz de detectar nos alvos) (VINHAS; SOUZA, 2005). A quantidade de regiões do espectro que de uma imagem possui implica na quantidade de bandas disponíveis, conforme representado na Figura 2.

Os dados espaciais de formato matricial mais utilizados atualmente são as imagens de satélite. Entretanto, existem outros tipos de dados que a representação computacional mais adequada é a matricial, como aquelas com fenômenos qualitativos contínuos, como altimetria obtida via RADAR (VINHAS; SOUZA, 2005). Uma das características mais importantes do dado matricial é a sua representação. No caso de imagens de satélite, a representação é feita com base no tamanho de cada pixel (cada elemento da matriz), o que é chamado de resolução espacial da imagem, que equivale à extensão nas direções horizontais e verticais de cada pixel (VINHAS; SOUZA, 2005).

2.2 Dados espaciais fuzzy

Ao longo dos anos houve uma grande adesão do uso da teoria e a lógica *fuzzy* em métodos de aplicações e análises espaciais na comunidade de geociência e SIG, o que resultou em uma grande quantidade publicações sobre abordagens do *fuzzy* no contexto de SIG (SCHNEIDER, 2008). Segundo Schneider (2008), o conceito espacial de vagueza ou *fuzzy* existe se um local não puder, completamente, ser associado ao conceito ou ao seu complemento. Estes objetos não podem ser representados por dados espaciais nítidos (*crisp*)

Figura 2 – Representação matricial multi-dimensional



Fonte: Adaptado de [Vinhas e Souza \(2005\)](#).

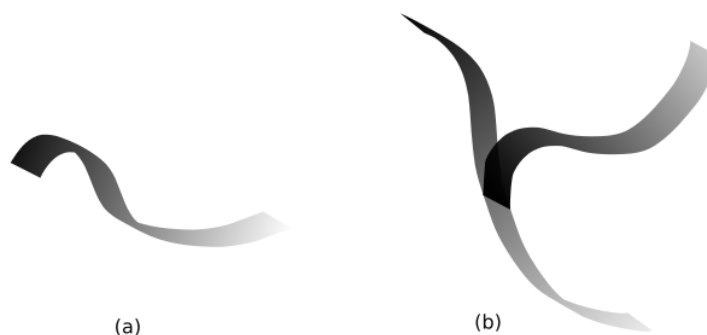
pois a sua característica de vagueza espacial não pode ser negligenciada ([CARNIEL, 2014](#)). Representações espaciais *fuzzy* são indicadas para demonstrar casos de transições, como tipos de vegetação, poluição do ar, temperatura, etc, uma vez que as regiões de borda não são capazes de demonstrar a transição gradual que ocorre no mundo real ([JASIEWICZ, 2011](#)). Esses objetos permitem flexibilizar o conceito de pertencimento ou não a uma determinada classe (0 e 1), substituindo pelo grau de pertinência, que pode variar dentro do intervalo de 0 a 1 ([CARNIEL; SCHNEIDER, 2018](#)).

Objetos espaciais *crisp* (ponto, linha e polígono) são consolidados na comunidade SIG e banco de dados espaciais, como abstrações apropriadas do mundo real bidimensional. Porém, suas representações como objetos espaciais vagos ainda é um conceito abstrato, baseado na teoria e topologia *fuzzy* ([SCHNEIDER, 2008](#)). A seguir são apresentados, de forma genérica, as definições dos objetos espaciais *fuzzy*, segundo [Schneider \(2008\)](#):

- Pontos *fuzzy*: Em um contexto *crisp*, pontos são elementos com localização x,y que podem pertencer ou não a um conjunto X . No contexto *fuzzy*, um ponto *fuzzy* também possui uma localização x,y porém ele pode pertencer a mais de um conjunto, ou seja, pertencer parcialmente a vários conjuntos.
- Linha *fuzzy*: As linhas são a abstração geométrica para recursos lineares como rios, limites e rotas de transporte. Cada elemento de uma única linha *crisp* é um ponto *crisp*

que definitivamente e totalmente pertencente à linha. Um objeto de linha *crisp* inclui um conjunto finito de blocos. Cada bloco consiste em um conjunto finito de linhas simples (curvas) de modo que cada par de linhas simples seja disjunto ou se encontre em um ponto final comum. As linhas *fuzzy* possuem a mesma estrutura das linhas *crisp*. Ou seja, uma linha *fuzzy* tem a mesma geometria linear que uma linha *crisp*, entretanto, cada ponto da linha *fuzzy* pode ter um grau de pertinência do conjunto X , quanto que a linha *crisp* pode apenas pertencer ou não pertencer ao conjunto X . A Figura 3 ilustra a representação espacial de uma linha *fuzzy*, na qual seu grau de pertinência é indicado pelas tonalidades mais intensas (grau próximo a 1) e mais suaves (grau próximo a 0) de cinza.

Figura 3 – Exemplos de linha *fuzzy*. a) linha *fuzzy* simples. a) linha *fuzzy* complexa

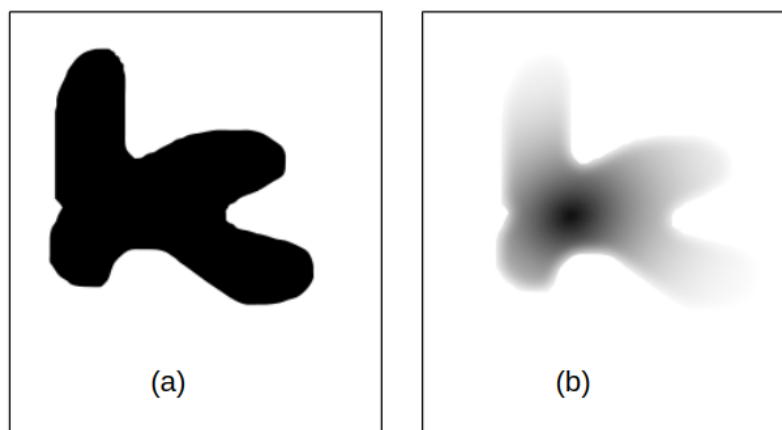


Fonte: Adaptado de [Schneider \(2008\)](#)

- Polígono (ou região) *fuzzy*: A estrutura dos polígonos *fuzzy* é semelhante à estrutura dos polígonos *crisp*, com a exceção de que o objeto *fuzzy* fornece uma flexibilidade em relação ao grau de pertinência à um conjunto ou uma classe. A Figura 4 ilustra uma comparação um polígono (ou região) *crisp* e um polígono (ou região) *fuzzy*.

O manuseio de dados espaciais são baseados exclusivamente na suposição de que um objeto espacial (ponto, linha ou região) é homogêneo, ou seja, todos os pontos em seu interior pertencem à uma mesma classe e suas bordas são perfeitamente delimitadas, isto é, baseia-se em objetos espaciais *crisp*. Entretanto, como já foi abordado nos tópicos anteriores, esse tipo de dado não representa a realidade em muitos casos. Para possibilitar a utilização de dados *fuzzy* em bancos de dados espaciais, e, conseqüentemente, possibilitar aplicações de álgebras espaciais *fuzzy* (FUSA), a utilização da Álgebra Espacial *Plateau* (*Spatial Plateau Algebra - SPA*) ([SCHNEIDER, 2014](#)) vem se mostrando promissora ([CARNIEL; SCHNEIDER, 2019](#)).

A *Spatial Plateau Algebra* (SPA) visa implementar, representar e aproximar computacionalmente objetos espaciais *fuzzy* em bibliotecas de geometria computacional existentes (feitos para representações *crisp*). O uso da SPA por [Schneider \(2014\)](#) inclui tipos de dados espaciais, operações e predicados de pontos plateau, linhas plateau e regiões *plateau* implementados em

Figura 4 – Exemplos de objeto espacial *fuzzy*. a) representação binária; b) representação *fuzzy*.

Fonte: Autora

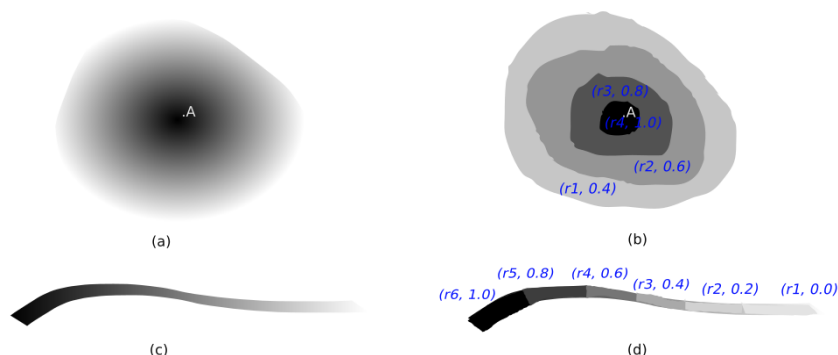
seu formato fuzzificado. A aproximação de objetos espaciais *fuzzy* e *plateau* possuem dois aspectos principais:

1. Por ser impossível representar um número infinito de graus de pertinência de um objeto espacial *fuzzy*, foram criadas classes de pertinência arbitrárias, nas quais devem representar, de forma finita, a quantidade de graus de pertinência de um objeto espacial *fuzzy*. Ou seja, todos os pontos que têm graus de pertinência próximos são agregados em grupos, e tais grupos assumem um único valor de grau de pertinência, que representa o conjunto de pontos agregados (SCHNEIDER, 2014).
2. Um objeto espacial *plateau* é uma coleção finita, onde cada objeto espacial nítido está associado a um valor de associação único e, portanto, forma um “platô”, plano consistindo em um número conceitualmente infinito de pontos de pertinência igual. A precisão com que representamos um objeto espacial *fuzzy* como um objeto espacial *plateau* depende do tamanho e do número de objetos componentes nítidos do objeto de platô espacial. Quanto menor o tamanho e maior o número de objetos componentes nítidos, melhor é a aproximação e maior é a precisão da representação de um objeto espacial *fuzzy* (SCHNEIDER, 2014).

Cada objeto espacial *plateau* consiste em uma coleção representável finita de objetos *crisp* e suas finitas contrapartes, que são adjacentes ou disjuntos entre si e rotulados em diferentes graus de pertinência. Tal coleção forma um conjunto *plateau* se considerados os graus de pertinência que indicam os graus de pertinência dos objetos espaciais *fuzzy* (CARNIEL; SCHNEIDER, 2019). A Figura 5 ilustra o conceito de representação de objeto espacial *fuzzy* como um objeto *plateau*.

O maior benefício da SPA é que sua implementação para o contexto *fuzzy* é baseada em modelos já existentes e implementados para dados espaciais do tipo *crisp* (CARNIEL; SCHNEIDER, 2019). Como exemplo de implementação da SPA, o pacote *fsr* (CARNIEL et al., 2021) possui um conjunto de funções para utilização de álgebra espacial *plateau* e

Figura 5 – a) exemplo de objeto espacial *fuzzy*. b) sua representação em formato de objeto *plateau*. c) representação de uma linha *fuzzy* e d) sua representação em formato de objeto *plateau*.



Fonte: Adaptado de [Schneider \(2014\)](#)

diferentes políticas de fuzzificação para criação de objetos espaciais *fuzzy*, baseado em ([CARNIEL; SCHNEIDER, 2019](#)).

2.3 Construção de objetos espaciais *fuzzy*

A construção de objetos espaciais *fuzzy* segue duas etapas. A primeira etapa consiste no uso de alguma política de fuzzificação para atribuir graus de pertinência (entre 0 e 1) ao conjunto de dados. Essa fase envolve a execução de algoritmos *fuzzy* de clusterização ou classificação. A segunda etapa consiste na aplicação de uma política de construção para obtenção de objeto espacial *plateau* com base nos graus de pertinência dos dados obtidos na etapa anterior.

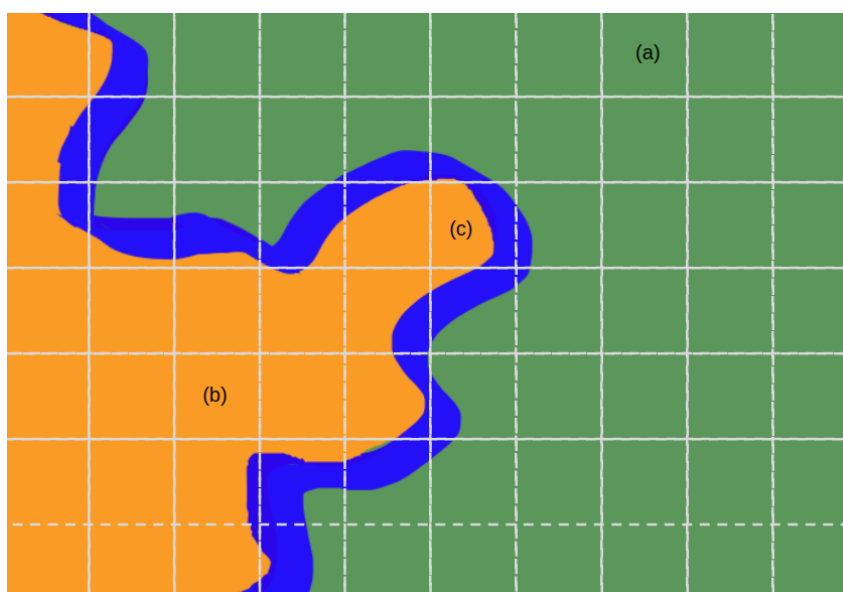
O pacote *fsr* implementa esse método de construção baseado em duas etapas por meio da função `spa_creator`, que gera objetos espaciais *fuzzy* a partir dos seguintes inputs: *tbl* - um conjunto de pontos com valores numéricos; *fuzzy_policy* - política de fuzzificação selecionada pelo usuário e *const_policy* - política de construção selecionada pelo usuário ([CARNIEL et al., 2021](#)).

2.4 Modelo de Mistura Espectral

Dados matriciais são compostos por pixels puros e pixels impuros. Tal pureza está relacionada com os diferentes materiais ou objetos contidos em um pixel (ou seja, em uma porção imaginária do terreno com dimensões definidas) no momento em que sua radiação foi capturada por um sensor e pode acontecer em dois casos. O primeiro caso ocorre quando os materiais (ou objetos) são menores que o pixel (neste caso o fluxo de radiação detectado pelo sensor é composto de uma mistura de radiação de todos os materiais dentro do pixel).

O segundo caso ocorre quando o pixel se sobrepõe à fronteira entre dois ou mais materiais ou objetos maiores do que ele. Em ambos os casos os sinais capturados pelo sensor não são representativos de qualquer um dos materiais presentes (SHIMABUKURO; PONZONI, 2017), conforme ilustrado na Figura 6.

Figura 6 – Representação de uma imagem com pixels misturados e puros: a) exemplo de pixel puro da classe representada pela cor verde na imagem; b) exemplo de pixel puro da classe representada pela cor laranja na imagem, e; c) exemplo de misturado com composição das classes verde, laranja e azul na imagem



Fonte: Adaptado de Shimabukuro e Ponzoni (2017).

A Figura 6 ilustra uma imagem formada por sete linhas e dez colunas (70 pixels), onde cada pixel está representado por linhas tracejadas e os objetos (ou materiais) são representados pelas cores na imagem. Observa-se que pixels representados por *a* e *b*, por exemplo, possuem apenas uma coloração sem seu interior, portanto são considerados pixels puros. Por outro lado, o pixel representado por *c* é considerado pixel misturado (impuro), pois possui mais de uma classe representada em seu interior (classes verde, laranja e azul). A complexidade da composição de um pixel é definida pelo termo mistura espectral, e existem vários modelos matemáticos propostos para identificação da proporção de mistura do pixel, sendo o Modelo Linear de Mistura Espectral (MLME), proposto por Shimabukuro e Smith (1991), uma das técnicas mais difundidas (FERREIRA, 2003).

O MLME visa identificar a contribuição que cada material tem na composição de um pixel, permitindo que a imagem seja analisada de forma 'suave' e precisa, na escala do subpixel. Neste contexto, o termo 'suave' remete ao conceito de *soft classification*, onde os pixels são analisados pela maior ou menor proporção de componentes existentes na mistura, e não pelo pertencimento a uma determinada classe temática (SHIMABUKURO; PONZONI, 2017).

Esta técnica é passível de ser utilizada a partir do conhecimento prévio da resposta

espectral dos materiais (ou objetos) que compõem o pixel. Uma das abordagens mais utilizadas para a aplicação do MLME é a aquisição de pixels puros (ou *endmembers*) disponíveis na própria imagem analisada (SHIMABUKURO; PONZONI, 2017; SILVÁN-CÁRDENAS; WANG, 2010). O MLME assume que a mistura espectral das imagens são resultantes de uma combinação linear das assinaturas espectrais dos diferentes materiais (*endmembers*) e a contribuição de cada material é, a princípio, proporcional à sua área de cobertura no pixel e a estimativa de cobertura relativa de cada material no pixel é obtida de forma linear (SILVÁN-CÁRDENAS; WANG, 2010). Considerando esta abordagem, presume-se que a partir do conhecimento das respostas espectrais de cada componente (*endmembers*), é possível estimar as proporções de cada componente da mistura (DUTRA; SHIMABUKURO; ARAI, 2018).

O conceito de MLME foi utilizado de diversas maneiras, com o surgimento de variantes aplicadas nos mais diversos campos. Um aplicação derivada do MLME é a análise de mistura espectral com múltiplos *endmembers*, conhecida como MESMA (*Multiple Endmember Spectral Mixture Analysis*), na qual visa corrigir algumas limitações observadas nos modelos lineares mais simples, que utilizam apenas um grupo de *endmembers* (ROBERTS et al., 1998).

O pacote RStoolbox (LEUTNER; HORNING; LEUTNER, 2017) possui uma implementação da MESMA em R. A função 'mesma' deste pacote possui a seguinte assinatura:

```
mesma(img, em, method = "NNLS", iterate = 400, ..., verbose)
```

Onde *img* é a imagem de satélite multiespectral (com várias bandas) em formato RasterStack ou RasterBrick (nativo do pacote *raster* (HIJMANS, 2022)), *em* é a matriz ou dataframe com os *endmembers*, *method* refere-se a um método de mistura espectral, *iterate* é a máxima interação por pixel, entre outros parâmetros possíveis de ajuste.

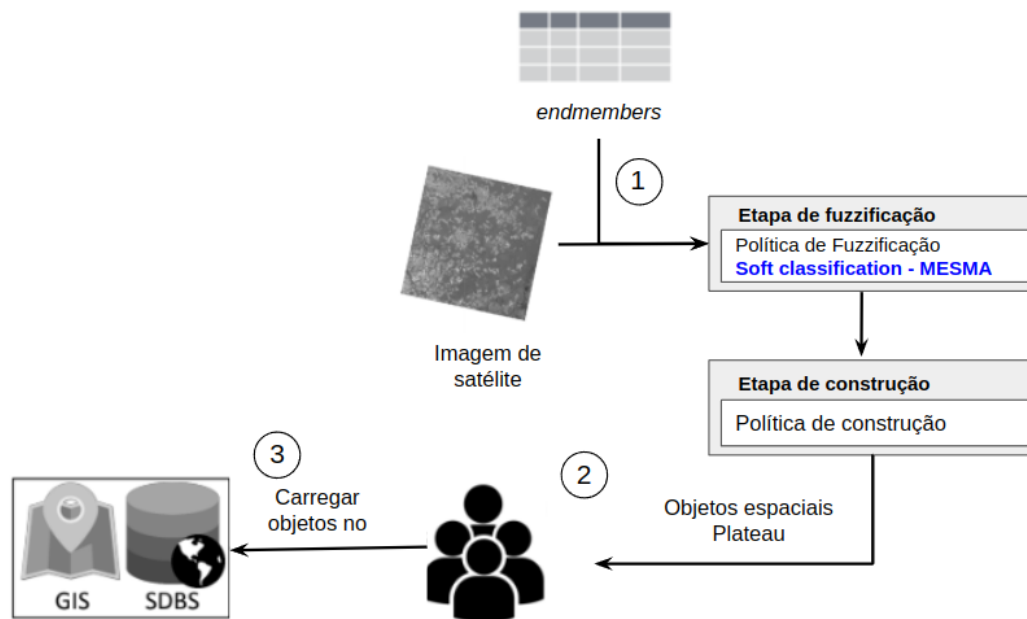
3 CONSTRUÇÃO DE OBJETOS ESPACIAIS FUZZY A PARTIR DE DADOS MATRICIAIS

Este capítulo detalha a abordagem sistemática para criação de objetos espaciais *fuzzy* a partir de dados espaciais matriciais em formato de imagens de satélite.

3.1 Visão Geral

A Figura 7 ilustra a visão geral de como uma nova política de fuzzificação pode ser incorporada ao procedimento de construção de objetos espaciais *fuzzy* no pacote *fsr*. Uma vez que o *fsr* é baseado na SPA, seus objetos espaciais *fuzzy* são ditos como *objetos espaciais plateau* (termo a ser empregado ao longo do capítulo).

Figura 7 – Visão geral da criação de objetos espaciais *plateau* a partir de imagens se satélite



Fonte: Adaptado de Carniel e Schneider (2019)

O primeiro passo para a construção de objetos espaciais *plateau* consiste na obtenção do *dataset* de entrada, que corresponde em uma imagem de satélite (formato raster .tif) e um *dataframe* com informações dos *endmembers* de cada uma das classes em cada uma das bandas da imagem. O segundo passo é dividido em duas etapas: 1) fuzzificação, onde é aplicada a política de fuzzificação no *dataset* de entrada e 2) política de construção, onde são aplicados algoritmos no resultado da fuzzificação, objetivando criar os objetos espaciais *plateau*. Por fim, o terceiro passo consiste na ingestão dos objetos espaciais *plateau* nos bancos de dados geográficos ou utilização em *softwares* SIG.

Classe	B2	B3	demais bandas
Vegetação	valor na banda B2	valor na banda B3	...
Não vegetação	valor na banda B2	valor na banda B3	...
Água	valor na banda B2	valor na banda B3	...

Tabela 1 – Estrutura de dataframe com valores de *endmembers* por banda espectral e pos classe

Como o objetivo deste trabalho é a incorporação de uma política de fuzzificação na construção de objetos espaciais *fuzzy*, as etapas 1 e 2 são detalhadas na Seção 3.2

3.2 Inclusão de uma nova política de fuzzificação

A primeira etapa (representada pelo número 1) na Figura 7 corresponde ao conjunto de dados de entrada, que consiste em uma imagem de satélite e um *dataframe* com os *endmembers* (valores de pixels puros) para cada banda espectral nas classes de interesse. Um exemplo de estrutura do *dataframe* é apresentado na Tabela 1. Os valores de *endmembers* podem ser obtidos em *softwares* ou plataformas de processamento digital de imagens (como o QGIS).

De posse do *dataframe* e da imagem de satélite correspondente, os próximos passos são as políticas de fuzzificação e construção, representados pelo número 2 na Figura 7. A etapa de fuzzificação neste contexto, como já abordado anteriormente, atribui valores de pertinência das classes aos pixels, de acordo com as suas respostas espectrais. Cada grau de pertinência indica o quanto aquele pixel pertence ou não à uma determinada classe, podendo variar de 0 a 1, onde 0 indica nenhum pertencimento e 1 indica o pertencimento total (caso de pixels puros).

O método de *soft classification* utilizado como política de fuzzificação é o MESMA, implementado pelo pacote *RStoolbox* (LEUTNER; HORNING; SCHWALB-WILLMANN, 2019). Este método foi utilizado para atribuir os graus de pertinência de cada pixel às classes de interesse. O algoritmo MESMA foi incorporado à função `mesma_policy` (política de fuzzificação proposta neste trabalho) que tem como entrada o caminho de uma imagem de satélite (caminho do diretório de uma imagem no formato .tif) e um dataframe de *endmembers* ('class_samples'). Como saída, a função gera um *dataframe* com coordenadas geográficas dos centróides de cada pixel com os níveis de pertinência por classe. O código da função é apresentado no Listing 3.1 mostrado a seguir.

```

1 library(RStoolbox)
2 library(raster)
3
4 mesma_policy <- function(path_to_image, class_samples) {
5
6   n_columns_dataframe <- ncol(class_samples)
7   img <- system.file(path_to_image, package = "raster")
8   img_brick <- brick(img)

```

```

9  nbands <- nbands(img_brick)
10
11  if (n_columns_dataframe == nbands) {
12    probs <- mesma(img_brick, class_samples, method = "NNLS")
13    probs_spatImg <- rast(probs)
14    data_fsr <- as.data.frame(probs_spatImg, xy = TRUE)
15
16    return (data_fsr)
17  } else {
18    print ("Number of bands is different from the dataframe columns
19          number")
20  }

```

Listing 3.1 – Função *mesma_policy*

A primeira etapa do código é a importação das bibliotecas necessárias para o funcionamento da função (linhas 1 e 2 ocorre a importação das bibliotecas *RStoolbox* e *raster* (HIJMANS, 2022)). Em seguida a função *mesma_policy* é declarada, e recebe como *inputs* o caminho da imagem de satélite (em formato 'string') e o *dataframe* com as informações de *endmembers* (linha 4). A função foi estruturada para receber a quantidade de colunas do *dataframe* igual à quantidade de bandas da imagem de entrada. Portanto, as quantidades de coluna do *dataframe* e a quantidade de bandas da imagem são comparadas com uma condicionante (linhas 6 a 11). Caso a quantidade de colunas do *dataframe* seja igual à quantidade de bandas da imagem, a função aplica a política de fuzzificação (linhas 12 a 16). Dentro da função *mesma_policy* a imagem de satélite é declarada dentro da variável 'img' por meio do pacote *raster* (linha 7) e convertida para o formato 'RasterBrick', nativo do pacote *raster* (linha 8). A função *mesma* (implementada no pacote *RStoolbox*) é aplicada nos *inputs* da função: imagem de entrada no formato 'RasterBrick', *dataframe* com *endmembers* e o método = 'NNLS' (único disponível no pacote) e armazenada na variável 'probs' (linha 12). O resultado da função é um *spatial raster* (linha 13), que é convertido para *dataframe* (linha 14) com as coordenadas de cada pixel da imagem com seus respectivos graus de pertinência para cada uma das classes declaradas no *dataframe* de entrada. Caso a quantidade de colunas do *dataframe* seja diferente da quantidade de bandas da imagem, a função retorna a seguinte mensagem: "Number of bands is different from the dataframe columns number".

O *dataframe* de saída da função serve como subsídio para construção de objetos espaciais *plateau*, por meio de políticas de construção da função já implementados pela função *spa_creator* do pacote *fsr*.

3.3 Incorporação da política no pacote *fsr*

O primeiro passo para a incorporação da função *mesma_policy* no pacote *fsr* foi a criação de uma *Issue* ('New fuzzification policy #32') no GitHub do pacote (<<https://github.com/robertiannunzi/FSR/issues/32>>).

([//github.com/accarniel/fsr](https://github.com/accarniel/fsr)>). Após esse passo e aplicação de melhorias no código, a função foi adicionada como política de fuzzificação no *builder_functions.R* do pacote, conforme mostrado no Listing 3.2.

```

1 #' mesma policy, which is based on the spectral mixture analysis
   implemented using the Multiple Endmember Spectral Mixture Analysis (
   MESMA) from the RStoolbox package.
2 #'
3 #' @param tbl A character vector to specify the full path of the
   satellite image in .tif format.
4 #' @param class_samples A data.frame with end members of the interest
   classes to be mapped. Obtained via image satellite. In the columns:
   first column: inters classes; other columns: band names (same as
   satellite image) and rows must present the pixel value by interest
   class.
5
6
7 #'
8 #' @return a tibble containing 'n' new attributes, where 'n' corresponds
   to the classes.
9 #'
10 #' @examples
11 #'
12 #' library(e1071)
13 #' set.seed(7)
14 #' tbl = tibble(x = runif(10, min= 0, max = 30), y = runif(10, min = 0,
   max = 50), z = runif(10, min = 0, max = 100))
15 #' fcp <- fuzzy_clustering_policy(tbl, 3)
16 #' fcp
17 #'
18 #' @importFrom RStoolbox mesma
19 #' @importFrom raster brick
20 #' @importFrom raster nbands
21 #' @importFrom raster as.data.frame
22 #' @import tibble
23 #' @noRd
24 mesma_policy <- function(tbl, class_samples, ...) {
25   if(!is.character(tbl)) {
26     stop("The path to the image must be a character vector", call. =
       FALSE)
27   }
28
29   # first, we need to read the image by using the package raster
30   img_brick <- brick(tbl)
31
32   n_columns_dataframe <- ncol(class_samples)
33   nbands <- nbands(img_brick)
34

```

```

35  if (n_columns_dataframe != nbands) {
36    stop("Number of bands is different from the number of the dataframe
      columns", call. = FALSE)
37  }
38
39  # second, we call the Multiple Endmember Spectral Mixture Analysis (
      MESMA) by using the package RStoolbox
40  # the mesma function has only one possible method (i.e., NNLS)
41  # mesma is one type of spectral mixture analysis
42  probs <- mesma(img_brick, class_samples, method = "NNLS")
43
44  # then, we return the result of the mesma as a tibble
45  result <- as_tibble(raster::as.data.frame(probs, xy = TRUE))
46  result <- result[, c(1, 2, ncol(result), 3:(ncol(result)-1))]
47
48  # we need to "regularize" the membership degrees
49  # all degrees greater than 1, will be 1
50  # all degrees lesser than 0, will be 0
51  result[, 4:ncol(result)][result[, 4:ncol(result)] > 1] <- 1
52  result[, 4:ncol(result)][result[, 4:ncol(result)] < 0] <- 0
53
54  result
55 }

```

Listing 3.2 – Função *mesma_policy* incorporada ao pacote *fsr*

A maior parte das modificações aplicadas no 3.2 é referente à padronização dos termos de acordo com os utilizados no pacote *fsr*, bem como a documentação dos parâmetros utilizados. Uma diferença de aplicação e funcionamento em relação ao código inicial é que esta nova versão da função *mesma_policy* realiza um tratamento adicional no conjunto de dados retornado pela função *as.data.frame* do pacote *raster* (linhas 43 a 52). Observou-se que a função *mesma_policy* retorna valores em intervalos variados entre 0 e 1, entretanto com o acréscimo do RMSE (*root mean square error*) nos valores finais, fazendo com que os valores retornados não sejam fixados no intervalo mínimo de 0 e máximo de 1, mas sim com uma pequena variação para maior ou menor. Esse RMSE é retornado pela função *mesma* como uma banda resultante da aplicação do MLME, sendo um importante dado para indicar a qualidade dos resultados do modelo de mistura espectral. Para fins da aplicação deste trabalho os valores de RMSE foram desconsiderados, mas podem ser utilizados futuramente para melhorar a qualidade desta política de fuzzificação. Portanto, foi aplicado uma aproximação dos valores menores que 0 e maiores que 1 para atenderem aos valores máximos aceitos em uma política de fuzzificação (linhas 43 a 52).

Como a função consiste em uma política de fuzzificação, ela foi incorporada de forma a ser utilizada na função *spa_creator*, conforme mostrado no Listing 3.3:

```

1 spa_creator <- function(tbl, fuzz_policy = "fsp", const_policy = "
      voronoi", ...) {

```

```

2  # should we validate the params here instead of validating them in the
    policies?
3  params <- list(...)
4
5  # first step is to apply the fuzzification step
6  fuzz_stage <- switch(fuzz_policy,
7    fsp = do.call(fuzzy_set_policy, c(list(tbl = tbl), params)),
8    fcp = do.call(fuzzy_clustering_policy, c(list(tbl = tbl), params)
9    ),
10     mesma = do.call(mesma_policy, c(list(tbl = tbl), params)),
11     stop(paste0("The fuzzification policy '",
12               fuzz_policy,
13               "' is not a supported policy.
14               The values are 'fsp', 'fsc', 'mesma'."),
15           call. = FALSE))
16 # from https://stat.ethz.ch/R-manual/R-devel/library/base/html/integer
    .html
17 is.wholenumber <- function(x, tol = .Machine$double.eps^0.5) abs(x -
    round(x)) < tol
18
19 if(hasArg("digits")) {
20   if(is.wholenumber(params$digits)) {
21     fuzz_stage[ , 4:ncol(fuzz_stage)] <- round(fuzz_stage[ , 4:ncol(
22     fuzz_stage)], params$digits)
23   } else {
24     stop("The argument 'digits' has to be an integer value.", call. =
25     FALSE)
26   }
27 }
28 }
29
30 ##### o codigo continua na versao original da funcao

```

Listing 3.3 – Função *mesma_policy* incorporada na função *spa_creator*.

Um exemplo da utilização da função *mesma_policy* usando uma chamada da função *spa_creator* é mostrado abaixo:

```

plateau <- spa_creator("/data/R-files/IMAGEM_SATELITE.tif", fuzz_policy
= "mesma", class_samples = endmembers, digits = 3)

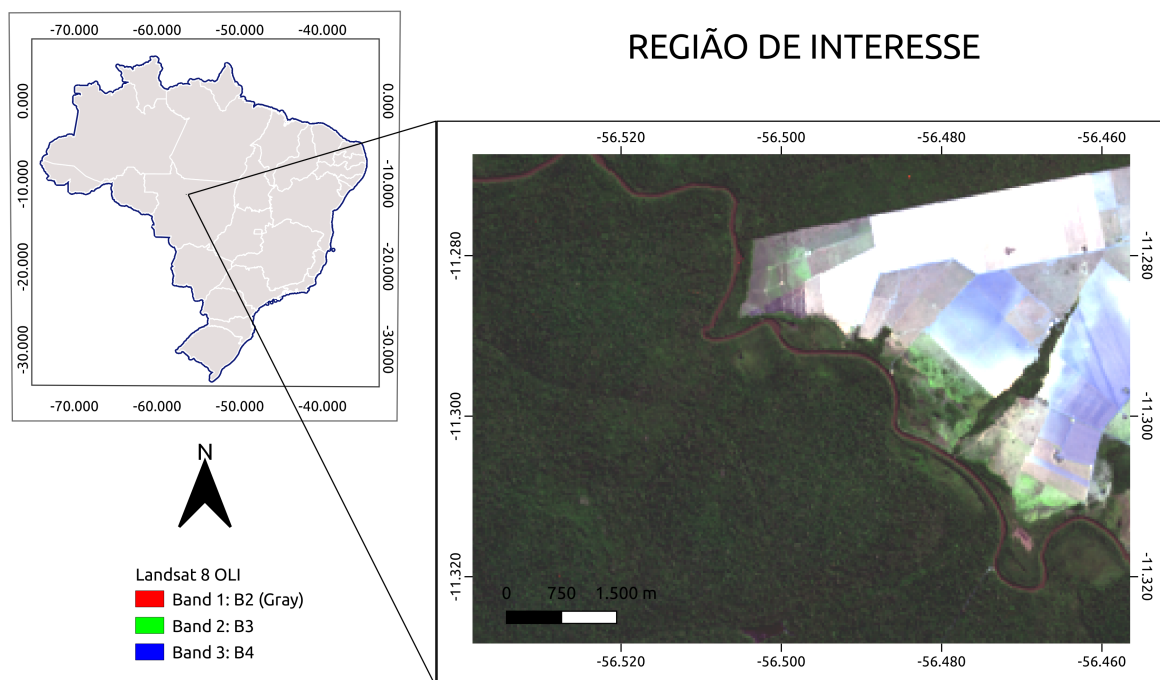
```


4 APLICANDO A NOVA POLÍTICA DE FUZZIFICAÇÃO EM UMA SITUAÇÃO REAL

4.1 Contexto da aplicação e obtenção das imagens de satélite

Com objetivo de ilustrar o resultado da política de fuzzificação proposta, uma imagem de satélite contendo três classes de cobertura do solo foi obtida: 'água', 'vegetação' e 'não vegetação'. A área para teste foi escolhida de forma que cada uma das classes mapeadas fossem bem representadas na Figura 8. A região está localizada no Estado de Mato Grosso, próximo ao município de Sinop, com as seguintes coordenadas geográficas: $-56^{\circ}30'0''$ W e $-11^{\circ}30'0''$ S em graus no DATUM EPSG: 4326 (*World Geodetic Survey de 1984*).

Figura 8 – Mapa de localização da área de interesse. Imagem Landsat 8 sensor OLI, composição 'cor verdadeira': bandas BLUE-GREEN-RED (B2B3B4) nos canais vermelho, verde e azul, respectivamente



Fonte: Autora

4.1.1 Obtenção da imagem de satélite

Atualmente, existem diversos tipos de imagens de satélite disponíveis tanto provenientes de iniciativas privadas quanto públicas. Essas imagens podem ser obtidas de diversos tipos de sensores, com diferentes resoluções espaciais, espectrais, radiométricas e temporais. Para a finalidade desta pesquisa, foram utilizadas imagens provenientes do satélite Landsat 8 OLI (disponibilizado pela U.S. Geological Survey). Trata-se de imagens de uso público com 30 metros de resolução espacial (resolução moderada), resolução temporal de 16 dias, radiométrica de 12 bits e espectral abrangendo as bandas do visível, infravermelho próximo, médio de ondas curtas e termal. Como as imagens são obtidas diariamente para o globo inteiro, foi necessário selecionar apenas aquelas que apresentam sobreposição com a área de interesse e selecionar apenas as bandas adequadas para utilização no estudo de caso, portanto foram obtidas apenas um recorte de imagens de uma cena e apenas do ano de 2021, com 359 pixels na horizontal e 238 pixels na vertical e as bandas espectrais: azul (B2), verde (B3), vermelho (B4), infravermelho de ondas curtas (B5), infravermelho médio 1 (B6) e infravermelho médio 2 (B7). Toda a fase de obtenção, filtragem e tratamento das imagens foram realizados em ambiente de computação em nuvem, por meio da plataforma Google Earth Engine (GORELICK, 2013) utilizando a API JavaScript. Tais imagens, após tratamento prévio, foram baixadas para trabalho em ambiente local.

O código em JavaScript utilizado para obtenção da imagem da região de estudo é mostrado no Listing 4.1.

```
1 \\Coordenadas geograficas da regio de interesse
2 var region_interest = ee.Geometry.Polygon(
3     [[[-56.51906572364731, -11.268651684202153],
4        [-56.51906572364731, -11.300973457688412],
5        [-56.48301683448715, -11.300973457688412],
6        [-56.48301683448715, -11.268651684202153]]], null, false);
7
8 Map.setCenter(-56.51906572364731, -11.268651684202153, 13);
9
10 \\ obtem colecao landsat para a regio de estudo, periodo e seleciona
    apenas as bandas de interesse.
11 var landsat = ee.ImageCollection("LANDSAT/LC08/C01/T1_TOA")
12     .filterDate('2021-07-01', '2021-08-01')
13     .filterBounds(region_interest)
14     .select(['B2', 'B3', 'B4', 'B5', 'B6', 'B7'])
15     .filterMetadata('CLOUD_COVER', 'less_than', 80)
16     .mean();
17
18 \\recorte da imagem apenas para a regio de interesse
19 var landsat1 = landsat.clip(region_interest)
```

Listing 4.1 – Código para obtenção da região de interesse em ambiente Google Earth Engine. Link para script: [<https://code.earthengine.google.com/>](https://code.earthengine.google.com/)

Classe	B2	B3	B4	B5	B6	B7
Vegetação	0.084	0.069	0.040	0.392	0.159	0.057
Não vegetação	0.333	0.344	0.379	0.388	0.538	0.497
Água	0.088	0.065	0.0444	0.051	0.021	0.009

Tabela 2 – Dataframe com valores de *endmembers* por banda espectral e pos classe referentes à localização do estudo de caso

[04e56adc64a3da3deb6b6d533ed9b388>](#)

4.1.2 Obtenção dos *endmembers*

De posse da imagem de interesse, a próxima etapa para obtenção dos *inputs* da função foi a obtenção dos *endmembers*. Como mencionado anteriormente, os *endmembers* nada mais são que as respostas espectrais de cada uma das bandas dos pixels puros em cada uma das classes de interesse ('água', 'vegetação' e 'não vegetação'), em formato de *dataframe*. Os pixels puros utilizados como *inputs* nesse estudo de caso foram coletados de forma manual diretamente na imagem de satélite obtida, por meio da utilização de algum *software* de SIG, como o QGIS. Importante ressaltar que a coleta desses *endmembers* é de extrema importância para o bom funcionamento do modelo, pois pixels com misturas de outras classes (ou seja, pixels impuros) podem resultar em resultados não satisfatórios. Recomenda-se, portanto, que a coleta desse dado seja feita por um intérprete previamente treinado. Os *endmembers* resultantes da coleta devem ser organizados em um *dataframe* que serve como *input* da função, conforme apresentado na Tabela 2.

No contexto deste trabalho foi coletado apenas um conjunto de *endmembers*, pois o objetivo é testar a implementação dessa nova política de fuzzificação. Entretanto, a função *mesma* do *RStoolbox* aceita múltiplos *endmembers*.

4.2 Aplicação do método de fuzzificação (aplicação da função *mesma_policy*) e política de construção

Uma vez obtidos os *inputs*, seguindo o fluxograma mostrado na Figura 7, a etapa seguinte é a aplicação da política de fuzzificação, que define o grau de pertinência de cada um dos pixels em relação às classes de entrada ('água', 'vegetação' e 'não vegetação'). Como resultado, foi obtido um *dataset* de 85.442 pontos (igual ao total de pixels da imagem - 359*238), contendo o grau de pertinência de cada pixel em relação a cada uma das classes de entrada, sendo as variações dos valores entre 0 e 1.

A próxima etapa consiste na política de construção dos objetos espaciais *plateau* a partir de uma política de construção *fuzzy* já implementada no pacote *fsr*. A política escolhida para construção foi a triangulação de Delaunay, que já está implementada na função *spa_creator*.

Como ambas as etapas foram implementadas na função *spa_creator*, o código utilizado para políticas de fuzzificação e construção é o mesmo. O Listing 4.2 apresenta o código utilizado na aplicação do estudo de caso.

```
1 class_names <- c("veg", "non_veg", "agua")
2
3 endmembers <- data.frame(B2 = c(0.0845, 0.333, 0.0889),
4                           B3 = c(0.069, 0.344, 0.065),
5                           B4 = c(0.040, 0.379, 0.044) ,
6                           B5 = c(0.392, 0.388, 0.051),
7                           B6 = c(0.159, 0.538, 0.021),
8                           B7 = c(0.057, 0.497, 0.009))
9
10 rownames(endmembers) <- class_names
11
12 path_to_image = 'PATH/TO/IMAGE.tif'
13
14 plateau_objs <- spa_creator(path_to_image, fuzz_policy = "mesma", class_
    samples = endmembers, digits = 3)
```

Listing 4.2 – Código utilizado no estudo de caso aplicando a política de fuzzificação *mesma_policy* incorporada na função *spa_creator*.

No Listing 4.2, o primeiro passo é a criação de um *dataframe* com os valores de *endmembers* coletados e mostrado na Tabela 2 (linhas 1 a 10). Em seguida é definido o caminho da imagem de satélite (linha 12). O último passo é executar a função *spa_creator* (linha 14), passando os parâmetros: *path_to_image* (caminho da imagem - linha 12); *fuzzy_policy* (política de fuzzificação) = 'mesma' (chama a função *mesma_policy*); *class_samples* = *test_endmembers* (*endmembers* coletados) e *digits* = 3 (quantidade de dígitos nas casas decimais)).

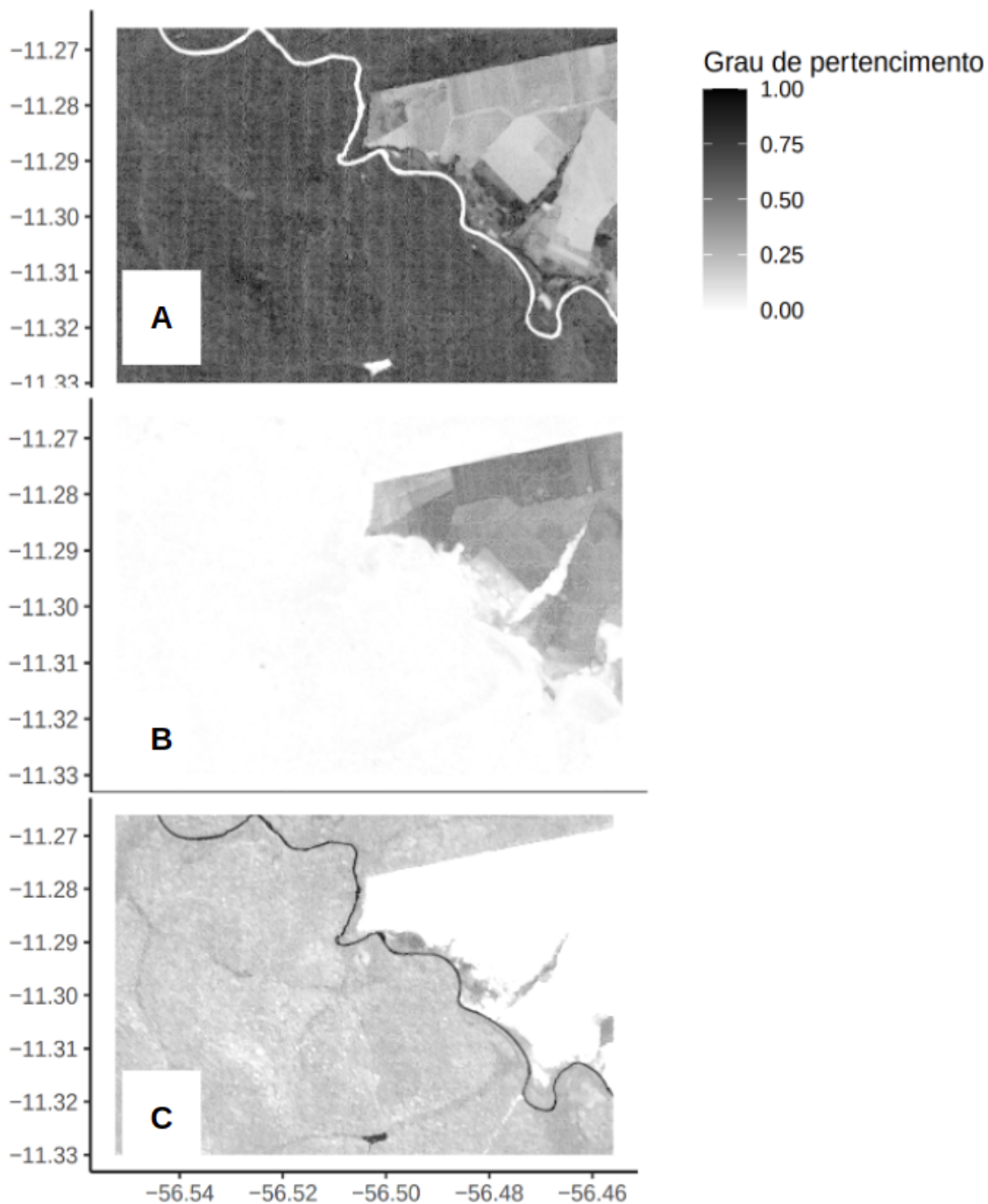
A Figura 9 apresenta objetos espaciais *plateau* resultantes da função *spa_creator*. A intensidade das cores representa o quanto o pixel pertence a uma determinada classe, sendo as cores mais escuras um indicativo de maior pertinência (valor 1) e cores mais claras representam menor pertinência (valor 0).

A figura resultante é bastante similar à figura de entrada, utilizada como *input* na função, porém nas regiões de transição, ou seja, nas regiões de borda, é possível notar regiões de transição na intensidade entre tons escuros e claros, conforme mostrado na Figura 10.

Na Figura 10 é possível notar algumas diferenças entre a imagem de satélite (*input*) e o resultante da função (objetos espaciais *plateau*). A principal diferença é o efeito de 'pixel' da imagem de satélite, que de fato é formada por esses pequenos elementos. No entanto, no objeto espacial *plateau* resultante da função *spa_creator* o efeito de pixel é atenuado e substituído por triangulações feitas na etapa de construção.

Outro fato interessante que vale ressaltar são os níveis de pertinência ilustrados nas porções C, D e E da Figura 10, que apresenta as diferentes intensidades nas tonalidades. A porção C da figura é referente ao nível de pertinência na porção água. Assim, pixels de

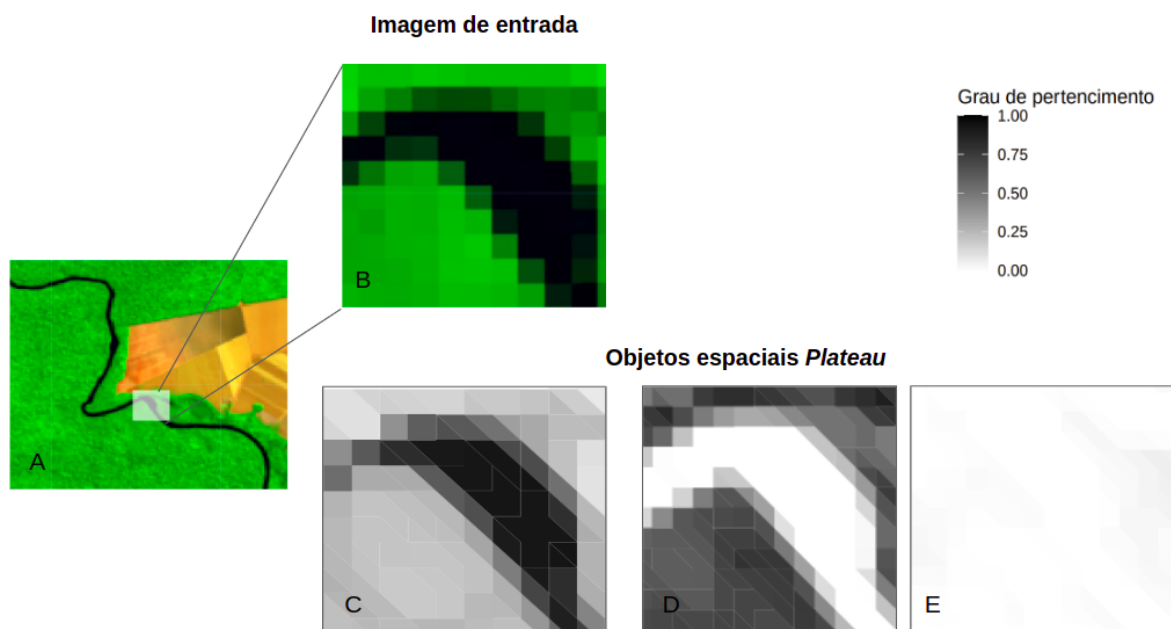
Figura 9 – Objetos espaciais *Plateau* resultantes da política de construção "mesma". A) Graus de pertinência da classe vegetação; B) graus de pertinência da classe de não vegetação e C) graus de pertinência da classe água.



Fonte: Autora

dentro do rio na imagem de satélite apresentam cores mais fortes, próximas ao preto. Por outro lado, na porção de vegetação as cores apresentam uma transição para o mais claro. Vale ressaltar que florestas possuem uma retenção de água. Dessa forma, é correto elas apresentarem

Figura 10 – Detalhamento dos resultados obtidos. A) imagem de satélite utilizada como *input*. B) detalhamento de uma região de água. C) resultado do objeto espacial *Plateau* para a classe água. D) resultado do objeto espacial *Plateau* para a classe vegetação. E) resultado do objeto espacial *Plateau* para a classe não vegetação.



Fonte: Autora

valores entre 0 e 1. De forma semelhante, ao analisar a porção D da imagem, na porção de vegetação, o rio apresenta coloração clara, enquanto que a porção da floresta apresenta cores mais escuras. A porção de não vegetação nesta parte da imagem é praticamente inexistente, conforme representado na porção E da figura.

5 CONCLUSÕES E TRABALHOS FUTUROS

Com o aumento do uso da lógica e teoria de conjuntos *fuzzy* no contexto da criação e análise de dados espaciais, existem várias iniciativas para implementação de políticas de fuzzificação em objetos espaciais vetoriais, como pontos, linhas e polígonos. Entretanto, existe uma lacuna em implementações deste tipo em dados matriciais, como imagens de satélite, que são uma fonte rica de dados especializados em escala global. Dentro desse contexto, este trabalho visou apresentar uma nova política de fuzzificação voltada ao uso de dados matriciais (formato *raster*), tendo em vista a carência de implementações *fuzzy* nesse tipo de dado.

O uso de classificadores do tipo *soft classification* mostrou-se satisfatório como política de fuzzificação. O método utilizado neste trabalho foi o Modelo Linear de Mistura Espectral, que atribui níveis de pureza aos pixels de uma imagem, dado um conjunto de pixels de referência como entrada do modelo (neste trabalho chamados de *endmembers* ou pixels puros). A partir deste modelo, foi criada uma função, denominada *mesma_policy*, que foi incorporada ao pacote *fsr* na função *spa_creator*. A aplicação do método apresentado em um estudo de caso envolvendo imagens de satélite mostrou resultados satisfatórios para o uso de *soft classification* como política de fuzzificação voltado a esse tipo de dado.

Tais resultados possibilitam vislumbrar trabalhos futuros no sentido da criação de novas políticas de fuzzificação voltadas à dados do tipo matriciais, e conseqüentemente, análise do desempenho destas políticas, uma vez que esse tipo de dado (*raster*) requer capacidade de processamento considerável.

Referências

- CÂMARA, G. Representação computacional de dados geográficos. **CASANOVA, MA et al. Banco de dados geográficos. Curitiba: Mundogeo**, p. 11–52, 2005. Citado na página 10.
- CARNIEL, A. C. Incorporando dados espaciais vagos em data warehouses geográficos: a proposta do tipo abstrato de dados vaguegeometry. Universidade Federal de São Carlos, 2014. Citado 3 vezes nas páginas 7, 8 e 12.
- CARNIEL, A. C. et al. Handling fuzzy spatial data in r using the fsr package. In: **ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems**. [S.l.: s.n.], 2021. p. 526–535. Citado 4 vezes nas páginas 7, 8, 14 e 15.
- CARNIEL, A. C.; SCHNEIDER, M. Spatial plateau algebra: An executable type system for fuzzy spatial data types. In: IEEE. **2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. [S.l.], 2018. p. 1–8. Citado 2 vezes nas páginas 7 e 12.
- CARNIEL, A. C.; SCHNEIDER, M. A systematic approach to creating fuzzy region objects from real spatial data sets. In: IEEE. **2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. [S.l.], 2019. p. 1–6. Citado 7 vezes nas páginas 7, 8, 11, 13, 14, 15 e 18.
- DILO, A.; BY, R. A. D.; STEIN, A. A system of types and operators for handling vague spatial objects. **International Journal of Geographical Information Science**, Taylor & Francis, v. 21, n. 4, p. 397–426, 2007. Citado 3 vezes nas páginas 7, 8 e 11.
- Aplicação do modelo linear de mistura espectral por abordagem combinada entre resoluções espaciais de imagens do sensor PROBA-V**. Citado na página 17.
- FERREIRA, M. E. Análise do modelo linear de mistura espectral na discriminação de fitofisionomias do parque nacional de Brasília (bioma cerrado). MSc Thesis, Universidade de Brasília, Brasília, 2003. Citado 2 vezes nas páginas 8 e 16.
- GORELICK, N. Google earth engine. In: AMERICAN GEOPHYSICAL UNION VIENNA, AUSTRIA. **EGU General Assembly Conference Abstracts**. [S.l.], 2013. v. 15, p. 11997. Citado na página 25.
- HIJMANS, R. J. **raster: Geographic Data Analysis and Modeling**. [S.l.], 2022. R package version 3.5-15. Disponível em: <<https://CRAN.R-project.org/package=raster>>. Citado 2 vezes nas páginas 17 e 20.
- JASIEWICZ, J. A new grass gis fuzzy inference system for massive data analysis. **Computers & Geosciences**, Elsevier, v. 37, n. 9, p. 1525–1531, 2011. Citado 2 vezes nas páginas 7 e 12.
- JENSEN, J. R.; EPIPHANIO, J. C. N. **Sensoriamento remoto do ambiente: uma perspectiva em recursos terrestres**. [S.l.]: Parêntese Editora São José dos Campos, 2009. Citado na página 11.
- LEUTNER, B.; HORNING, N.; LEUTNER, M. B. Package ‘rstoobox’. **R Foundation for Statistical Computing, Version 0.1**, 2017. Citado na página 17.

- LEUTNER, B.; HORNING, N.; SCHWALB-WILLMANN, J. **RStoolbox: Tools for Remote Sensing Data Analysis**. [S.l.], 2019. R package version 0.2.6. Disponível em: <<https://CRAN.R-project.org/package=RStoolbox>>. Citado na página 19.
- LIU, B. et al. A model of fuzzy topological relations for simple spatial objects in gis. **Boletim de Ciências Geodésicas**, SciELO Brasil, v. 21, p. 389–408, 2015. Citado na página 7.
- RENZHONG, G. Spatial objects and spatial relationships. **Geo-spatial Information Science**, Taylor & Francis, v. 1, n. 1, p. 38–42, 1998. Citado na página 10.
- ROBERTS, D. A. et al. Mapping chaparral in the santa monica mountains using multiple endmember spectral mixture models. **Remote sensing of environment**, Elsevier, v. 65, n. 3, p. 267–279, 1998. Citado na página 17.
- SCHNEIDER, M. Fuzzy spatial data types for spatial uncertainty management in databases. In: **Handbook of research on fuzzy information processing in databases**. [S.l.]: IGI Global, 2008. p. 490–515. Citado 3 vezes nas páginas 11, 12 e 13.
- SCHNEIDER, M. Spatial plateau algebra for implementing fuzzy spatial objects in databases and gis: Spatial plateau data types and operations. **Applied Soft Computing**, Elsevier, v. 16, p. 148–170, 2014. Citado 3 vezes nas páginas 13, 14 e 15.
- SCHNEIDER, M.; BEHR, T. Topological relationships between complex spatial objects. **ACM Transactions on Database Systems**. Citado na página 10.
- SHIMABUKURO, Y. E.; PONZONI, F. J. **Mistura Espectral: modelo linear e aplicações**. [S.l.]: Oficina de Textos, 2017. Citado 3 vezes nas páginas 8, 16 e 17.
- SHIMABUKURO, Y. E.; SMITH, J. A. The least-squares mixing models to generate fraction images derived from remote sensing multispectral data. **IEEE Transactions on Geoscience and Remote sensing**, IEEE, v. 29, n. 1, p. 16–20, 1991. Citado na página 16.
- SILVÁN-CÁRDENAS, J. L.; WANG, L. Fully constrained linear spectral unmixing: Analytic solution using fuzzy sets. **IEEE Transactions on Geoscience and Remote Sensing**, IEEE, v. 48, n. 11, p. 3992–4002, 2010. Citado na página 17.
- VINHAS, L.; SOUZA, R. C. M. de. Tratamento de dados matriciais na terralib. 2005. Citado 2 vezes nas páginas 11 e 12.