

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

DIOGO OLIVEIRA NASCIMENTO

**Uma proposta para modificação
complementar de geometrias em sistemas
de versionamento de dados geoespaciais**

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em Ciência da
Computação

Orientador: Prof. Dra. Renata Galante

Porto Alegre
2022

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Nascimento, Diogo Oliveira

Uma proposta para modificação complementar de geometrias em sistemas de versionamento de dados geoespaciais / Diogo Oliveira Nascimento. – Porto Alegre: PPGC da UFRGS, 2022.

125 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2022. Orientador: Renata Galante.

1. Banco de dados geográficos. 2. Dados geoespaciais. 3. Versionamento de dados. 4. Sistema de Informações Geográficas. I. Galante, Renata. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões

Vice-Reitora: Prof^a. Patricia Pranke

Pró-Reitora de Pós-Graduação: Prof. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do PPGC: Prof. Dr. Claudio Rosito Jung

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

“Onde há vontade, há um caminho”

– AUTOR DESCONHECIDO

RESUMO

Atualmente, o processo de produção de dados geográficos é feito utilizando bancos de dados geográficos. Durante esse processo, existe a necessidade de coleta de dados em campo, no qual não há conexão com o banco de dados central. Assim sendo, diversas versões do conjunto de dados são modificadas em paralelo, a semelhança ao desenvolvimento de software por múltiplos desenvolvedores utilizando *Distributed Version Control System* (DCVS). Porém, os sistemas atuais não lidam com a geometria de maneira especial, de modo que o uso de dados geográficos em sistemas de versionamento de dados acaba introduzindo conflitos desnecessários. A presente dissertação tem por objetivo estabelecer um método que permite considerar modificações em diferentes subpartes de uma geometria durante o processo de *merge* dos *branches*, reduzindo assim possíveis conflitos de geometria. Para formalizar a proposta são definidas e ilustradas as diversas operações de GIS e também seus comportamentos em sistemas de versionamento, sendo também parte da contribuição da dissertação. O foco do trabalho é introduzido o conceito de *Modificação complementar de geometria* e seu método de aplicação para a integração de geometrias. A modificação complementar de geometria estabelece a definição de modificação de segmentos da geometria, a partir do alinhamento de vértices, bem como também estabelece a integração dos segmentos alterados para promover a integração das versões da geometria. São apresentados exemplos que demonstram a aplicação do método tanto em geometria do tipo linha quanto em área, mesmo quando ocorrem mudanças significativas na geometria como *flip* ou rotação de índices de *rings*. Para avaliar a proposta é realizado um estudo de caso em parceria com 1º Centro de Geoinformação, uma das unidades do Exército Brasileiro que produz dados geográficos para o Exército Brasileiro e para a sociedade civil. Foram usados dados oriundos de projetos realizados pela unidade militar para avaliar qualitativamente a proposta. Os voluntários especialistas evidenciaram que a *modificação complementar de geometria* resolve grande parte dos conflitos e que a proposta favorece a adoção de sistemas de versionamento em ambiente de produção de dados geográficos.

Palavras-chave: Banco de dados geográficos. Dados geoespaciais. Versionamento de dados. Sistema de Informações Geográficas.

A proposal of complementary geometry changes in a geographic data versioning system

ABSTRACT

Nowadays, the process of producing geographic data is done using geographic databases. There is a need to collect data in the field during this process, where there is no connection to the central database. So several versions of the dataset are modified in parallel, similar to software development by multiple developers using Distributed Version Control System (DCVS). However, current systems do not deal with geometry in a particular way, and versioning systems introduce unnecessary conflicts in geographic data. The present dissertation aims to establish a method that allows considering modifications in different subparts of geometry during the merge of the branches, thus reducing possible geometry conflicts. GIS operations and their behavior in versioning systems are defined and illustrated to formalize the proposal, which is also part of the contribution of this dissertation. As a focus of the work, the concept of *Complementary geometry changes* and its application method for the integration of geometries is introduced. Complementary geometry changes establish the definition of modification of segments of the geometry from the alignment of vertices and establish the integration of the changed segments to promote the integration of the geometry versions. Examples are presented that demonstrate the application of the method in both line and area geometry, even when there are significant changes in geometry such as *flip* or rotation of *rings* indices. A case study is carried out in partnership with 1º Centro de Geoinformação, one of the units of the Brazilian Army that produces geographic data for the Brazilian Army and for civil society, to evaluate the proposal. Data from projects carried out by the military unit were used to evaluate the proposal qualitatively. The expert volunteers showed that *complementary geometry changes* resolves most of the conflicts and that the proposal favors the adoption of versioning systems in a geographic data production environment.

Keywords: Geographic databases, Geospatial data, Data versioning, Geographical Information System.

LISTA DE ABREVIATURAS E SIGLAS

1º CGEO	1º Centro de Geoinformação, da DSG
ACID	<i>Atomicity, Consistency, Isolation, Durability</i>
CAD	<i>Computer-aided design</i>
CVCS	<i>Centralized Version Control System</i>
DAG	<i>Directed acyclic graph</i>
DCVS	<i>Distributed Version Control System</i>
DSG	<i>Diretoria de Serviço Geográfico, do Exército Brasileiro</i>
GIS	<i>Geographic Information System</i>
ET-EDGV	Especificação Técnica para Estruturação de Dados Geoespaciais Vetoriais
IBGE	Instituto Brasileiro de Geografia e Estatística
OGC	<i>The Open Geospatial Consortium</i>
OGC - SFA	<i>OCG - Simple Feature Access</i>
PLC	<i>Programmable logic controllers</i>
SCN	Sistema Cartográfico Nacional
SDT	<i>Spacial Data Types</i>
SID	<i>Spatial Identifier</i>
SQL	<i>Structured Query Language</i>
UUID	<i>Universally Unique Identifier</i>
VCS	<i>Version Control System</i>
WC	<i>Working Copy</i>
WKT	<i>Well-known Text</i>
WKB	<i>Well-known Binary</i>

LISTA DE FIGURAS

2.1	Representação visual dos WKT.....	17
4.1	Operações para geometrias do tipo ponto.....	35
4.2	Operações básicas para geometrias do tipo linha	37
4.3	Operação de <i>flip</i> para geometrias do tipo linha	37
4.4	Operações compostas para geometrias do tipo linha.....	39
4.5	Operações básicas para geometrias do tipo área.....	41
4.6	Operações compostas para geometrias do tipo área	43
4.7	Alterações indiretas em geometrias do tipo área	44
4.8	Casos de <i>insert</i> de geometrias do tipo ponto	49
4.9	Casos de <i>delete</i> de geometrias do tipo ponto.....	50
4.10	Casos de <i>update</i> de geometrias do tipo ponto	50
4.11	Casos de <i>insert</i> de geometrias do tipo linha	52
4.12	Casos de <i>delete</i> de geometrias do tipo linha.....	53
4.13	Casos de <i>update</i> por <i>insert</i> de vértices em geometrias do tipo linha.....	54
4.14	Casos de <i>update</i> por <i>delete</i> de vértice em geometrias do tipo linha.....	56
4.15	Casos de <i>update</i> por <i>flip</i> de geometrias do tipo linha	57
4.16	Casos de <i>update</i> por movimentação de vértice em geometrias do tipo linha.	58
4.17	Casos de <i>update</i> por <i>cut</i> de geometrias do tipo linha	59
4.18	Casos de <i>update</i> por <i>merge</i> de geometrias do tipo linha	60
4.19	Casos de <i>insert</i> de geometrias do tipo área.....	62
4.20	Casos de <i>delete</i> de geometrias do tipo área	63
4.21	Casos de <i>insert</i> de <i>hole</i> em geometrias do tipo área.....	64
4.22	Casos de <i>update</i> por <i>delete</i> de <i>hole</i> em geometrias do tipo área	65
4.23	Casos de <i>update</i> por <i>insert</i> de vértice em geometrias do tipo área.....	66
4.24	Casos de <i>update</i> por <i>delete</i> de vértice em geometrias do tipo área	68
4.25	Casos de <i>update</i> por movimentação de vértices em geometrias do tipo área.	70
4.26	Casos de <i>update</i> por <i>cut</i> em geometrias do tipo área	71
4.27	Casos de <i>update</i> por <i>merge</i> de geometrias do tipo área.....	72
4.28	Casos de <i>update</i> por iteração no ordenamento de <i>hole</i> em geometrias do tipo área	73
4.29	Casos de <i>update</i> por rotação de índice em geometrias do tipo área	75
4.30	Casos de <i>update</i> por <i>flip</i> em geometrias do tipo área	76
5.1	Identificação de <i>insert</i> de vértice na comparação de vetores entre <i>branches</i> .	81
5.2	Identificação de <i>delete</i> de vértice na comparação de vetores entre <i>branches</i>	82
5.3	Identificação de operações compostas em linhas na comparação de vetores entre <i>branches</i>	83
5.4	Comparação de geometrias entre B^A , B^1 e B^2 para geometrias do tipo linha	85
5.5	Representação espacial do Exemplo 1.....	87
5.6	Representação do alinhamento dos vértices do Caso 1	87
5.7	Geometria integrada - Caso 1	90
5.8	Operação de <i>flip</i>	90
5.9	Representação espacial do Exemplo 2.....	92
5.10	Representação espacial dos Sub-casos do Exemplo 2.....	94
5.11	Exemplo de pareamento inicial de <i>hole</i> em geometrias áreas	96
5.12	Exemplo de pareamento de <i>hole</i> em geometrias áreas	97
5.13	Representação espacial do Exemplo 3.....	100

5.14	Pareamento de <i>holes</i> das versões B^A e B^1 da geometria A1 do Exemplo 3..	101
5.15	Pareamento de <i>holes</i> das versões B^A e B^2 da geometria A1 do Exemplo 3..	102
5.16	Geometria A1 integrada.....	104
6.1	Exemplo de geometria versionada.....	110
6.2	Exemplo de geometria integrada	112
A.1	Exemplo de geometria versionada.....	123
A.2	Exemplo de geometria integrada	124

LISTA DE TABELAS

3.1	Comparativo entre sistemas de versionamento de dados geográficos disponíveis.....	32
5.1	Exemplo de tabela de pareamento de <i>holes</i>	97
5.2	Exemplo 3 - tabela de pareamento de <i>holes</i> entre B^A e B^1	101
5.3	Exemplo 3 - tabela de pareamento de <i>holes</i> entre B^A e B^2	102

SUMÁRIO

1	Introdução	12
2	Fundamentação teórica	16
2.1	Conceitos gerais sobre dados geográficos	16
2.2	Bancos de dados espaciais	18
2.3	Versionamento	18
2.4	Considerações	23
3	Trabalhos relacionados	25
3.1	Trabalhos acadêmicos	25
3.2	Softwares Relacionados	27
3.2.1	GitSpatial	28
3.2.2	GeoGig	28
3.2.3	Kart	29
3.2.4	GeoDiff + Mergin	30
3.2.5	Análise Comparativa	31
4	Definições Preliminares	33
4.1	Operação de GIS sobre geometrias	33
4.1.1	Operações para camadas do tipo ponto	34
4.1.2	Operações para camadas do tipo linha	35
4.1.3	Operações para camadas do tipo área	38
4.2	Dados Geográficos em Sistemas de Versionamento de Dados	46
4.2.1	Versionamento para camadas de geometrias do tipo ponto	47
4.2.2	Versionamento para camadas do tipo linha	49
4.2.3	Versionamento para camadas de geometrias do tipo área	61
4.2.4	Conceito geral de versionamento aplicado a dados geográficos	76
5	Modificação complementar de geometria	78
5.1	Modificação complementar em geometrias	78
5.2	Modificação complementar em geometrias do tipo linha	79
5.2.1	Determinando as modificações entre geometrias do tipo linha	79
5.2.1.1	<i>Update</i> por <i>insert</i> de vértice	80
5.2.1.2	<i>Update</i> por <i>delete</i> de vértices	81
5.2.2	<i>Update</i> por operações compostas	83
5.2.3	Aplicação de modificação complementar em geometrias do tipo linha	84
5.2.4	Exemplo - Caso 1	86
5.2.5	Operações do tipo <i>flip</i> em geometrias do tipo linha	89
5.2.5.1	Determinando operação de <i>flip</i>	89
5.2.5.2	Exemplo - Caso 2	91
5.2.5.3	Operação de <i>flip</i> em modificação complementar de ge- ometrias	91
5.3	Modificação complementar em geometrias do tipo área	95
5.3.1	Determinando as modificações entre geometrias área	95
5.3.2	Aplicação de modificação complementar de geometrias do tipo área	98
5.3.3	Exemplo - Caso 3	99

5.3.4	Operações do tipo <i>flip</i> e rotação de índices em geometrias do tipo área.....	105
5.3.4.1	Identificando alteração do tipo <i>flip</i> e seu tratamento durante a modificação complementar de geometrias.....	105
5.3.4.2	Identificando alteração do tipo rotação de índices e seu tratamento durante a modificação complementar de geometrias	105
5.4	Considerações finais sobre a proposta	106
6	Estudo de Caso - 1º Centro de Geoinformação	107
6.1	Metodologia.....	107
6.2	Resultados.....	109
6.2.1	Demografia.....	109
6.2.2	Descrição das Tarefa e Análise dos Resultados	109
6.3	Análise Geral dos Resultados	114
6.3.1	Limitações.....	115
7	Conclusão	116
	Referências	120
	APÊNDICE A — Questionário de avaliação do método	122
A.1	Informações pessoais	122
A.2	Familiarização com versionamento de dados geográficos e a resolução de conflitos de geometria.....	123
A.3	Conceito de modificação complementar de geometria.....	123
A.4	Teste do método em um conjunto de dados.....	124

1 INTRODUÇÃO

O avanço da digitalização dos processos do mundo físico permitiu que diversas atividades antes executadas somente do modo analógico pudessem ter existência ou processos equivalentes no meio digital. Essa digitalização impactou muitas profissões e ofícios, dentre eles, muitos dos trabalhos relacionados com as Engenharias. Esse processo de transformação impactou particularmente a Engenharia Cartográfica, tendo em vista que no atual estágio das tecnologias boa parte das atividades executadas pelos Engenheiros Cartógrafos são realizadas com o auxílio de um computador e, mais que isso, muitas das análises hoje possíveis só são viáveis pela forma de trabalho em meio digital. A construção e o uso das base de dados geográficos é feita com o uso de Sistemas de Informações Geográficas (GIS - *Geographic Information System*), que em sentido mais amplo são a combinação de *hardware* e *software* com profissionais para a coleta, armazenamento, gerenciamento, manipulação, com a produção de modelos e análises para auxílio no processo de tomada de decisão e planejamento que usem informações geográficas (WADEMBERE; OGAO, 2010).

Para armazenar os dados coletados, diversos formatos de dados geográficos surgiram, porém para permitir o intercâmbio de dados entre diversos sistemas, houve a necessidade de se estabelecer padrões. Entre os diversos padrões de dados se destacam os padrões da *The Open Geospatial Consortium* - (OGC), organização que reúne os principais fabricantes de *software* geográficos e GIS, bem como governos e associações de usuários. Os padrões OGC definiram entre outros, o *Well-known Text* - WKT e o *Well-known Binary* - WKB, que são formas de armazenar os dados geográficos amplamente difundidos e usados na comunidade. O WKT facilita a leitura por parte dos usuários, assim facilitando o entendimento do que se trata a geometria, enquanto que o WKB é a variação do dado geográfico para binário, permitindo maior ganho quando armazenado, pois reduz o uso de espaço em memória (OGC, 2011).

Com a evolução dos sistemas, os GIS incorporaram o uso de banco de dados como repositório de dados, permitindo a partir desse momento o ganho das propriedades ACID (*Atomicity, Consistency, Isolation, Durability*) no gerenciamento dos dados geográficos. De modo a prover funções além do armazenamento do dado geográfico, foram incorporadas novos tipos de dados (dados geográficos), bem como novas funções (que permitem o uso das coordenadas armazenadas) para assim, permitir de fato, que o banco de dados usado tenha características especiais, tornando-se um banco de dados geográficos (GÜ-

TING, 1994). Um exemplo de banco de dados amplamente utilizado pela comunidade geoespacial, principalmente a comunidade relacionada com software livre, é o banco de dados Postgres com a extensão espacial PostGIS.

Com o uso de banco de dados geográficos, o GIS permite que muitos usuários se conectem a um único repositório de dados e trabalhem juntos na elaboração de novas bases e na complementação das bases já existentes. No processo de trabalho de uma linha de produção de dados para a construção de bases de dados geográficos uma das fases a ser executada é a de confirmação em campo, no qual os especialistas vão no terreno confirmar dados das feições que foram construídas na sede. Dessa forma, ainda que atualmente a principal parte dos processos seja executada em escritório com os operadores conectados em banco de dados central, existe uma fase em que os mesmos precisam executar atividades fora da sede. Pelo modo de trabalho desenvolvido, essa fase também é feita de forma digital em um *laptop* com o mesmo GIS da sede, no qual o operador complementa as informações que são colhidas em campo. Devido a precária rede de dados disponível em diversos locais do país, não existe a possibilidade dos operadores estarem conectados a base de dados original. Dessa forma, os mesmos vão a campo com bancos locais instalados em seus equipamentos de trabalho e executam os trabalhos necessários em suas cópias locais. Devido a essa atividade, existe a necessidade de posteriormente integrar todas as bases de dados, integrando os dados novos levantados pelos diversos operadores de campo.

Esse tipo de necessidade tem bastante semelhança com os DVCS (*Distributed Version Control System*) usados amplamente na elaboração de código de software. Os sistemas do tipo DVCS permitem que múltiplos usuários trabalhem *off-line* e que os trabalhos sejam posteriormente sincronizados em uma base (KOC; TANSEL, 2011). O fato do DVCS ser um sistema distribuído permite interação dos repositórios de modo não linear e não sequencial, o que permite múltiplas possibilidades de integração dos dados, não ficando limitado somente aos sistemas de repositório central, apesar desse ser o principal caso de uso. Um sistema que hoje se encontra em franca expansão de uso e que representa muito bem os DVCS é o GIT¹, que é amplamente usado pela comunidade de desenvolvedores de software. Assim como a área de desenvolvimento de software ganhou muito em produtividade e qualidade por uso de sistemas DVCS, é vislumbrado que a área de construção de base de dados geoespaciais, e a comunidade de dados geográficos como um todo, também tem muito a ganhar ao conseguir ter a possibilidade de usar sistemas

¹Sistema de versionamento de código descentralizado <<https://git-scm.com/>>

DVCS adaptado às necessidades e particularidades dos dados envolvidos em seu trabalho.

Existem soluções de versionamento para banco de dados em geral, mas são poucas as soluções que apresentam propostas para banco de dados espaciais (SANFORD, 2013; LOCATIONTECH, 2014; KOORDINATES, 2021). Dentre estas poucas, não existe uma solução que contemple a área de dados geográficos de modo integral. Muito pouco foi explorado no tocante às particularidades sobre os dados espaciais, principalmente no tocante à integração das geometrias versionadas. No modo atual de trabalho dos sistemas de versionamento de dados, muitas das interações acabam por resultar em conflitos que são oriundos do não aproveitamento das potencialidades do dado geográfico. O conflito em versionamento demanda mais trabalho manual por parte dos operadores para sua resolução e isso acaba por diminuir o uso dos sistemas de versionamento pela comunidade. Sendo assim, o presente trabalho visa contribuir nesse sentido, apresentando uma nova visão na possibilidade de resolução de conflitos para otimização da integração de dados espaciais em DVCS aplicado a banco de dados geográficos.

O **objetivo geral** desta dissertação é especificar uma nova proposta para modificação complementar de geometrias em sistemas de versionamento de dados geoespaciais. A nova proposta permite reduzir o conflito de integração de dados geoespaciais em sistemas de versionamento, por meio da integração de geometrias de versões paralelas de uma mesma base de dados. Inicialmente, são sistematizadas e formalizadas as operações de inserção, exclusão e atualização para as operações para ponto, linha e área para que possam ser executadas sobre geometrias. O conceito de versionamento é, então, especificado e formalizado para as camadas de geometria do tipo ponto, linha e área. Em seguida, a proposta de modificação complementar de geometrias é minuciosamente especificada para conduzir o processo de alteração em geometrias do tipo linha e área dentro de um contexto de integração de dados. A proposta foi qualitativamente avaliada através de um estudo de caso realizado no 1º Centro de Geoinformação do Exército Brasileiro através da análise de especialistas que atuam na coordenação da elaboração de base de dados geográficos. Resultados mostraram que é possível reduzir o esforço de resolução de conflitos no contexto de integração de diferentes versões do mesmo banco de dados geográfico.

As principais contribuições deste trabalho:

- sistematização das operações de inserção, exclusão e atualização para dados geográficos do tipo ponto, linha e área;
- especificação do conceito de versionamento para as camadas do tipo linha e área;

- proposição de uma nova proposta de modificação complementar de geometria e sua aplicação em versionamento de dados geoespaciais;
- a discussão da proposta apresentada com relação a sua adoção em ambiente de produção de dados geográficos.

Este trabalho está estruturado, a contar desta introdução, em seis capítulos:

- o Capítulo 2 apresenta os conceitos que fundamentam o trabalho, iniciando por conceitos gerais de dados geográficos, passando por banco de dados espaciais e pontuando alguns conceitos de versionamento;
- o Capítulo 3 apresenta os principais trabalhos relacionados ao tema, bem como os sistemas de versionamento atualmente disponíveis para dados geográficos;
- o Capítulo 4 sistematiza e formaliza as definições preliminares que embasam a proposta, introduzindo as operações de GIS, além de definir a maneira como os sistemas de versionamento atuam ao lidar com a geometria;
- o Capítulo 5 apresenta o conceito de *modificação complementar de geometria* caracterizando sua aplicação para os dados geográficos vetoriais, além da exemplificação para operação que promovem mudanças mais profundas na geometria, como as operações de *flip*, alteração de ordenamento de holes e rotação de índice de *rings*;
- o Capítulo 6 apresenta um estudo de caso para avaliação qualitativa da proposta feita com especialistas que atuam na coordenação da elaboração de base de dados geográficos;
- por fim, o Capítulo 7 explicita as conclusões, contribuições e trabalhos futuros resultantes desta pesquisa;
- O Apêndice A apresenta o tipo de questionário aplicado aos voluntários do estudo de caso.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos relacionados com esta dissertação, organizando-se em quatro seções. A Seção 2.1 apresenta alguns dos conceitos gerais para dados geográficos, para ambientação na parte geoespacial. A Seção 2.2 apresenta o que é banco de dados espaciais. Na seção 2.3, é discutido o tema de versionamento, e seus principais conceitos. Finalmente, na seção 2.4 são realizadas considerações finais sobre o capítulo.

2.1 Conceitos gerais sobre dados geográficos

Os tipos geométricos que são usados na construção de base de dados de mapeamento são os pontos, as linhas e as áreas. Os pontos são conjuntos de coordenadas, geralmente definidos por suas componentes (X Y), referenciadas a um sistema de coordenadas. As linhas e as áreas são compostas por pontos internos, que são mais conhecidos como vértices, porém eles têm a mesma composição de coordenadas dos pontos. As linhas são encadeamento de vértices, sendo compostas por no mínimo dois vértices distintos. As áreas, ou polígonos, são regiões internas demarcadas por no mínimo três vértices distintos, podendo ainda ter furos internos, os *hole* (SAMPAIO; BRANDALIZE, 2018). Os pontos / vértices podem ter a coordenada Z e ainda uma outra componente chamada de Measurement Dimension (M). Existem ainda outros tipos de dados, como as multi-geometrias, as superfícies, as *Geometry Collection*, sendo estas compostas pelos tipos geométricos apresentados anteriormente (OGC, 2011).

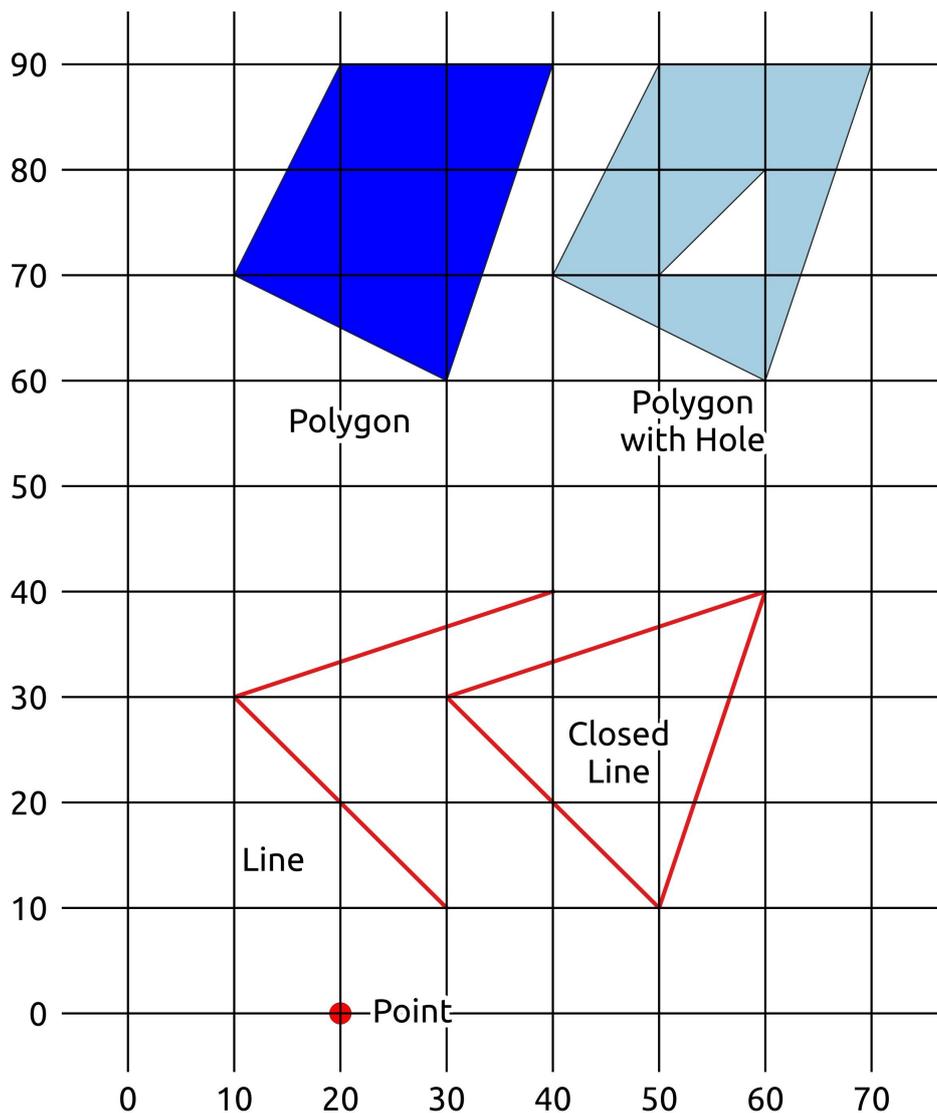
Para expressar um dado geográfico, é necessário definir como o mesmo deve ser interpretado pelo sistema. De modo a permitir a interoperabilidade entre sistemas, foram definidos padrões de dados geográficos, e entre os mais aceitos estão os propostos pela *Open Geospatial Consortium* (OGC). A OGC é uma organização que reúne os principais fabricantes de *software* geográficos e GIS, bem como governos e associações de usuários, e busca definir padrões consensuais livres a serem adotados por todos os envolvidos. A OGC em sua norma *Simple Feature Access* (OGC, 2011) define os padrões *Well-known Text* - WKT e o *Well-known Binary* - WKB. O WKT é um padrão de mais fácil compreensão humana e permite sua leitura de modo mais fácil para o usuário. Já o WKB é um padrão binário que permite ganhos na economia de espaço de armazenamento. A seguir, são apresentados alguns exemplos de geometrias expressas em WKT, e a Figura 2.1 traz

a representação visual das mesmas geometrias.

Exemplos de geometrias expressas em WKT:

- Ponto = POINT (20 0)
- Linha = LINESTRING (30 10, 10 30, 40 40)
- Linha fechada = LINESTRING (50 10, 30 30, 60 40, 50 10)
- Polígono = POLYGON ((30 60, 40 90, 20 90, 10 70, 30 60))
- Polígono com *hole* = POLYGON ((60 60, 70 90, 50 90, 40 70, 60 60),(50 70, 60 80, 60 70, 50 70))

Figura 2.1: Representação visual dos WKT



2.2 Bancos de dados espaciais

Banco de dados (ELMASRI et al., 2005) é uma coleção de dados relacionados (tabelas, relações) que possuem as seguintes propriedades implícitas: (i) representa aspectos do mundo real (mini-mundo ou universo de discurso); (ii) é uma coleção de dados logicamente coerentes com algum significado inerente; e (iii) é projetado, construído e instanciado (“povoado”) para uma aplicação específica. Um sistema de gerência de banco de dados (SGBD) é um conjunto de programas que permite criar e manter (manipular) um banco de dados (ELMASRI et al., 2005). Os bancos de dados tipicamente estão organizados para trabalhar com dados naturais do mundo, como números e texto, e para proporcionar melhor relação de armazenamento e acesso aos dados, são construídos tipos de dados que conseguem otimizar essas ações. Assim, são definidos tipos de dados como int8, int16, float, char(32), varchar, entre outros.

Os bancos de dados espaciais são bancos de dados que tem por objetivo gerenciar de forma efetiva e eficiente dados que representam o mundo físico (dados geográficos, planejamento urbano, dados astronômicos), partes de organismos vivos, projetos de engenharia (grandes escalas de circuitos integrados, desenhos de automóveis, estrutura molecular de uma droga farmacêutica) e informações conceituais do espaço (sistema multidimensional de suporte a decisão, dinâmica de fluídos, ou campo magnético) (SHEKHAR et al., 1999). Segundo Güting (1994), um banco de dados espacial é um banco de dados que oferece tipos de dados espaciais (*Spacial Data Types - SDT*) em seu modelo e na sua linguagem de consulta, e ainda provê no mínimo índice espacial e algoritmos eficientes para junções espaciais. Um Banco de Dados Espacial permite ainda que se tenha, em seu modelo de dados, tipos capazes de suportar estruturas que referenciam objetos do mundo físico e que, além da existência de índices que aceleram a busca de modo espacial dos dados, tenha-se também algoritmos que permitam o uso eficiente da componente espacial para fazer relacionamento entre as geometrias.

2.3 Versionamento

O tema de versionamento, ou melhor *Version Control System - VCS*, é um tema que foi amplamente explorado pela área de engenharia de software. Os primeiros trabalhos a respeito do tema remontam a década de 1970 e o seu principal propósito era permitir a colaboração entre desenvolvedores (KOC; TANSEL, 2011). VCS também pode

ser referido como *Revision Control System*, *Software Configuration Management*, *Source Code Management* ou ainda *Source Code Control*. Como sugerem os nomes alternativos, o VCS é bastante difundido na comunidade de desenvolvimento de software, para gerenciamento, controle e colaboração no desenvolvimento do código do software.

Baseados em Koc and Tansel (2011) e Chacon and Straub (2014), os principais conceitos e categorizações sobre VCS são apresentados a seguir. Os primeiros conceitos necessários para entender VCS são a definição de suas partes. Assim, pode-se estabelecer que um VCS é dividido em três partes:

- *working copy* (WC) é a estrutura dentro do sistema onde serão feitas as modificações pelos usuários;
- *Staging area* é a estrutura onde são colocadas as modificações para serem analisadas e posteriormente inseridas nas versões;
- O repositório é a estrutura onde são armazenadas as versões e as informações sobre as versões.

Em um VCS funcionando corretamente, após uma versão salva, é possível apagar a WC sem nenhuma perda ao sistema, já que a WC serve somente para o usuário realizar as modificações. Assim, após a WC apagada é possível recriá-la a partir dos dados armazenados no repositório. Contudo, apagar alguma das outras partes pode destruir todo o sistema. Por exemplo, apagar o repositório acaba por destruir o histórico dos dados gerenciados pelo sistema, uma vez que a WC é somente um estado dos dados e não possibilita reconstruir o histórico dos dados. Ainda sobre as estruturas, pode-se categorizar o repositório em local, quando armazenado na máquina do usuário; e repositório remoto, quando armazenado em servidores para interação de múltiplos usuários.

As operações são características importantes em sistemas de versionamento. As primeiras ações a serem apresentadas são diretamente relacionadas com as partes do sistema:

- *checkout*, ação de colocar uma WC em igualdade com uma versão específica armazenada no repositório;
- *add*, ação de uma vez feitas as alterações necessárias na WC, enviar as alterações para a *staging area*;
- *commit*, ação de pegar as informações da *staging area* e criar uma versão dos dados armazenando-a no repositório.

O próximo conjunto de operações é usado para acomodar as evoluções em trabalhos concomitantes, mais comum em sistemas distribuídos:

- *branch*, que é uma cópia feita a partir de uma determinada versão, e que agora não possui mais ligação com a evolução daquela versão em particular, permitindo que as duas evoluam de modo paralelo;
- *merge*, que é a junção de caminhos paralelos que tenham algum ponto comum de evolução, executadas para integrar trabalhos concomitantes executados em *branches* distintos;
- *clone*, que é a cópia de um repositório de um local para outro, em geral, cópia do repositório remoto para o repositório local;
- *push*, que é o envio das alterações armazenadas no repositório local para o repositório remoto, a fim de disponibilizar as alterações executadas pelo usuário a outros elementos que tem acesso ao repositório remoto;
- *pull*, que é a busca das novas alterações armazenadas no repositório remoto, trazendo-as para o repositório local, tornando os dois repositórios compatíveis.

Outro conceito a ser explorado é o *Atomic Data Unit*. Esse conceito é expresso pelo quão fino é o controle do VCS sobre o dado a ser gerenciado pelo sistema. *Atomic Data Unit* é a unidade que o sistema entende como sendo atômica, uma mudança em parte dessa unidade é uma mudança na unidade como um todo. Por exemplo, pode ser um texto ou arquivo como um todo, ou podem ser suas divisões, como capítulos, parágrafos, frases ou até palavras. Os sistemas atuais para versionamento de código em geral usam linhas do código (*statements*) como *atomic data unit*.

Outro conceito é relacionado com as estratégias de concorrência, principalmente quanto a alterações / escrita nos arquivos. Os VCS surgiram primeiro visando permitir a colaboração de diversos atores em um mesmo arquivo. Para que esse objetivo fosse alcançado era necessário estabelecer as estratégias de concorrência no acesso ao arquivo ou partes dele. Basicamente são duas que combinadas formam uma terceira: o *Lock*, o *Merge* e o Híbrido. Na estratégia de *Lock*, é necessário que um usuário estabeleça quais arquivos ele quer bloquear (em alguns sistemas era possível bloquear somente partes de um arquivos). Esse bloqueio não permite que os demais usuários do sistema alterem esses arquivos enquanto o bloqueio persistir. Esse tipo de estratégia é criticada, pois não resolve de fato a concorrência, já que os vários usuários têm que bloquear e desbloquear

os arquivos o que não permite o acesso concorrente para trabalho. A outra estratégia é o *Merge*, o que permite que cada usuário altere o arquivo de acordo com a sua necessidade e posteriormente faça a integração das partes não conflitantes, apontando ao usuário os trechos no código onde não foram possíveis fazer a integração das diversas versões de forma automatizada. Esta estratégia ganhou muitos adeptos, pois permite maior agilidade nos trabalhos, bem como maior flexibilidade. No entanto, se houverem muitas mudanças no código, esta estratégia pode gerar a necessidade de lidar com muitos conflitos, o que pode reduzir a eficiência do processo. A última estratégia é chamada de Híbrida, pois combina as duas anteriores. A estratégia híbrida usa cada uma em partes do processo, a depender da configuração, como, por exemplo, pode ser usado *Merge* em repositórios remotos e *Lock* em repositórios locais ou compartilhados.

Outro conceito a ser abordado é do tipo de repositório a ser usado pelo sistema. Inicialmente os VCS eram utilizados em sistemas de pastas compartilhadas, porém esse tipo de estrutura não era suficientemente ágil para trabalhar com muitos usuários. Assim, surgiu o formato de CVCS - *Centralized Version Control System* que estabelecia que para usar o VCS o usuário deveria estar conectado ao sistema que se encontrava em um servidor. Essa estrutura de Cliente / Servidor exigia que as alterações só fossem possíveis quando o usuário estivesse conectado, gerando a necessidade de conexão constante, o que desagradava os desenvolvedores, e às vezes tornava o trabalho improdutivo a depender da qualidade da conexão disponível. Por esses e outros problemas, a evolução do sistema caminhou para o DVCS - *Distributed Version Control System*, que não dependiam mais de uma conexão constante de cliente/servidor. Um DVCS, como o GIT (GIT, 2005), permite que o usuário faça uma cópia local do repositório, altere os arquivos de modo *off-line* e posteriormente compartilhe suas alterações com outros nós da rede composta pelos desenvolvedores. Dessa forma, o DVCS permite a evolução do código por meio de interações que formam um grafo que permite interações não sequenciais, gerando modos de trabalhos mais diversos do que somente a interação com um repositório central (apesar dessa ainda ser a forma mais comum de desenvolvimento, e que também é suportada pelo DVCS).

Há ainda o conceito do modo como são armazenadas as versões dos dados. A forma mais simples são *snapshots*, que são cópias integrais dos dados no momento do *commit*. Esse modo apresenta a vantagem de rapidez de acesso a qualquer versão dos dados, porém apresentam a desvantagem de aumentar muito rapidamente o espaço requerido para o repositório. Outro modo é por uso dos chamados *delta*, que é a diferença

necessária para que uma versão se iguale a outra. Os deltas são estabelecidos por uso de algoritmos de *diff*, no qual são identificadas as alterações feitas a partir da versão já armazenada para que esta se iguale a versão que está em uso, e ainda são estabelecidos *patches* que são as próprias alterações realizadas. Quando feito no sentido da versão armazenada para a versão corrente, é dito um delta direto, e quando feito no sentido da versão corrente, estabelecendo as alterações feitas na direção da versão armazenada, é dito como um delta inverso ou delta reverso. O delta reverso apresenta a vantagem de necessitar somente de uma versão completa dos dados, a que está em uso, e construir as versões anteriores pela aplicação dos deltas reversos.

Outro ponto importante a ser mencionado é quanto a forma de identificação de conflitos. Os sistemas atuais usam o modelo chamado *3 way*, no qual os possíveis trechos de conflitos são comparados entre as duas versões de interesse e também com a última versão comum entre elas. Para melhor explicar o modelo *3 way*, serão tomadas três versões estabelecidas no repositório, uma versão anterior comum (V^A), um *branch* a partir de V^A , chamado V^1 , e outro *branch* a partir de V^A , chamado de V^2 , porém sem ligação com a V^1 . Quando comparados V^1 e V^2 , alguns trechos não são mais iguais e podem ter três possíveis caminhos: uma evolução de V^1 com base V^A na qual V^2 ainda se encontra igual a V^A ; uma evolução de V^2 com base V^A na qual V^1 ainda se encontra igual a V^A ; uma evolução de V^1 e V^2 , para códigos diferentes entre si, e também com os dois diferentes de V^A . Desses três caminhos, somente o último é verdadeiramente um conflito, e só é possível estabelecer essa definição com base em V^A . No caso do modelo *2 Way*, a identificação de reais conflitos se torna impossível, pois a falta de registro do ponto comum entre as versões acarreta que mesmo trechos de código que tenham apenas evoluído em um dos *branches* se apresente como um conflito. O *2 way* foi um dos principais problemas enfrentados pelos primeiros sistemas VCS.

Por último, são apresentados os conceitos relacionados a *partial clone*, que são conceitos que permitem a implementação de funções no sistema de modo a facilitar o usuário a lidar com a complexidade crescente do repositório de um VCS, permitindo filtragem do repositório a ser clonado. Primeiro conceito a ser abordado é o conceito de *shallow clone*, que é um tipo de filtragem do repositório baseado na profundidade, isto é, número de versões anteriores, que se quer ter na clonagem. Esse tipo de filtragem permite que o sistema somente clone as últimas N versões armazenadas no repositório, tomando como base uma versão específica e as $N-1$ versões anteriores. Tal tipo de filtragem ajuda a reduzir o tamanho do repositório a ser clonado, o que pode ser muito útil, principal-

mente em projetos que tenha um histórico muito grande, e no qual os novos usuários não necessitem de acesso as versões mais antigas que a profundidade definida. O segundo conceito a ser abordado é o de *sparse clone*. Esse tipo de filtragem já se apresenta de modo mais generalista, onde podem ser feitas especificações de filtragem de dados dentro de uma versão particular. Um dos casos documentados com relação ao DVCS GIT é a possibilidade de não clonar arquivos binários, ou ainda, limitá-los a um certo tamanho X . A aplicação direta desses conceitos ao banco de dados pode permitir uma filtragem com base em determinadas classes de dados, que se materializam em tabelas no banco. Porém, para uso com dados espaciais essa filtragem de dados dentro de versões apresenta um uso muito vantajoso: filtragem por área espacial. Quando se está trabalhando com dados geográficos, os dados têm a componente espacial, o que é natural, porém um diferencial. Essa característica combinada com a forma como os dados são trabalhados durante sua produção e uso faz com que sempre se tenha uma área de interesse para o trabalho, independente de como esta área é definida. Pode ser a área do município de Porto Alegre, ou do estado do Rio Grande do Sul, ou o Brasil inteiro, ou ainda um certo bioma, ou bacia hidrográfica, podendo também ser definida como um enquadramento dentro de latitudes e longitudes. Todas essas porções de terreno, umas com caracterização geométrica mais fácil como divisão politico-administrativas, outras podendo ser definidas com certa aproximação, podem ser usadas como filtro espacial dos dados. Assim, o uso de filtro, tanto espacial como de classe de dados, permite que do repositório só sejam clonados os dados necessários ao usuário que irá trabalhá-los.

2.4 Considerações

O presente capítulo apresentou diversos conceitos relacionados ao tema e que são base para o desenvolvimento do trabalho. Foram apresentadas as especificações da OCG, e suas definições para WKT e WKB. Foi ainda introduzido o conceito de banco de dados geográficos, além de diversos conceitos relacionados a versionamento. Esses conceitos são parte do conhecimento necessário para que se possa desenvolver o raciocínio para apresentação da proposta de modificação complementar de geometrias. Foi feita a opção de uso de representação de geometria como WKT, pois é um tipo de representação usado e aceito pelos atores da comunidade de dados geográficos, e a proposta aqui apresentada em WKT pode ser depois adaptada para o tipo de dado particular que cada solução venha a usar. A opção por banco de dados se dá por ser um tipo de repositório de dados

extremamente confiável, além de utilizado por diversos atores da comunidade. Quanto as configurações do sistema de versionamento, a primeira escolha é pelo sistema do tipo DVCS, pois este comporta interações entre o *branches* paralelos de modos diversos e isso permite maior flexibilidade para o sistema. A segunda escolha é em relação ao tópico do *Atomic Data Unit*, que no caso será um dos pontos onde a proposta irá apresentar parte de sua contribuição ao permitir trabalhar com subpartes da geometria, em um processo que será explicado ao longo do trabalho. A terceira escolha importante é com relação ao modelo *3 way* de identificação de conflito visto que o mesmo já se provou superior ao modelo anterior.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta os trabalhos relacionados que balizam a presente pesquisa, tanto em âmbito acadêmico como em relação aos software disponíveis para uso. Muitos dos trabalhos apresentados aqui são analisados com base nas contribuições a respeito de versionamento de dados geográficos, particularmente a respeito da integração de versões oriundas de uma mesma base de dados. Nem todos os trabalhos contemplam todos os aspectos necessários, porém os artigos e softwares focados no exato tema são bem escassos e, por isso, os escolhidos são apresentados mesmo não contemplando todas as necessidades levantadas. O presente capítulo se encontra dividido em três seções: a primeira, a Seção 3.1, apresenta trabalhos acadêmicos que tratam sobre o tema; a segunda, a Seção 3.2, apresenta os *softwares* que atuam no seguimento ou atuaram, já que alguns deles se encontram descontinuados; por fim, a Seção 3.2.5 compara os trabalhos e softwares apresentados e apresenta os pontos em abertos contemplados pela proposta descrita nesta dissertação.

3.1 Trabalhos acadêmicos

Uma das formas encontradas na literatura que referenciam o versionamento de vetores de dados geoespaciais é por uso de identificador espacial, com o uso de identificador único atrelado a componente espacial, um *Spatial Identifier* - SID, como em (FELUSAB; SREBROA; TALA, 2010). O método proposto consiste em um identificador que é atribuído ao registro e que usa a sua posição espacial para sua definição. No caso do trabalho citado, é proposto um SID para as feições composto por quatro letras e um número (*long integer* - 32 bits), sendo as letras usadas para identificar a base de dados ao qual pertencem as feições e o número para identificar a feição dentro da base de dados. Apesar da proposta de uso para dados geográficos esse tipo de identificador possui restrições para uso em sistemas que fazem alterações na geometria, já que o identificador se baseia na componente geométrica do registro.

Outras pesquisas encontradas na literatura sugerem a inserção de um conjunto de tabelas, atributos e relacionamentos na modelagem da base de dados para que possam construir o versionamento, como em (LI et al., 2010). Tal proposição é focada em ter agilidade para trabalhar com múltiplas versões, já que não estabelece um conjunto de tabelas de versão corrente e trabalha com todas as tuplas de várias versões em uma única

tabela, permitindo construir uma versão específica por meio de *joins* e *where conditions* de SQL. O método é focado em agilizar a análise comparativa entre duas versões distintas dos dados. Outros trabalhos focam no melhoramento do algoritmo que identifica a diferença entre os tipos de dados. Por exemplo, no artigo (SVEEN, 2020), o autor apresenta os algoritmos de *diff* usados em contexto geoespacial que são algoritmos genéricos de *diff* combinados (pré- e pós-processamento) e propõem o *GeomDiff*, um algoritmo de *diff* e um formato de armazenamento do mesmo, voltados para dados geoespaciais. Apesar dos trabalhos serem focados em tópicos de versionamento de dados geográficos, nenhum deles apresenta propostas para melhorar a integração e *merge* das geometrias que estão sendo versionadas.

Há ainda outros trabalhos que trazem o enfoque de apresentar casos de uso de versionamento de dados geográficos, porém não foi encontrado nenhum caso com semelhança ao foco do presente trabalho. Dentre os diversos trabalhos podem ser apresentados o Khudyakov et al. (2018), no qual é escolhido o sistema de versionamento *Subversion*¹ para versionamento de arquivos CAD (*computer-aided design*) de projetos PLC (*programmable logic controllers*). Porém, arquivos de CAD são binários, e por isso o uso do SVN não permite usufruir de toda a potencialidade do sistema, limitando seu uso ao ordenamento de versões. Ainda de acordo com o trabalho foram usados os metadados dos *commits* para indicar as alterações entre versões. Tal proposta apresenta a desvantagem de necessitar a atenção rigorosa do usuário ao elencar as alterações nos *commits*, sendo tal fato uma desvantagem no modelo proposto pelos autores.

Outros trabalhos apresentam uso para tipos geométricos distintos. O trabalho de Le et al. (2014) explora a implementação de um sistema de versionamento voltado a dados geológicos, com base em estruturas do tipo *surface* definidas por triangulação de vértices. O sistema é montado para facilitar as comparações entre estruturas e permite obter o histórico, conseguindo reconstruir a estrutura usada para uma determinada análise. Ao reconstruir a estrutura permite estabelecer responsabilidades por eventuais análises deficientes, que podem impactar em diversos pontos da cadeia de eventos ocorridos, como no caso de pagamento de seguros em eventuais danos ocorridos. O tipo de dado usado no sistema e a forma de versionamento não se aplicam ao caso da presente dissertação.

Outras pesquisas apresentam enfoque em versionamento baseado em operações. No trabalho Cserép and Giachetta (2015), as operações são as alterações feitas nos dados. O sistema de versionamento guarda as informações detalhadas dessas operações. Por

¹Sistema de versionamento de código centralizado <<https://subversion.apache.org/>>

exemplo, quais foram os algoritmos aplicados e parâmetros definidos nas configurações dos algoritmos. Porém, este modelo é muito adequado a dados *raster*, onde um algoritmo é aplicado ao conjunto completo dos dados a serem trabalhados, não sendo essa a forma de se trabalhar com geometrias de dados vetoriais.

Como último tópico de trabalhos relativos ao tema a serem apresentados, tem-se os artigos Medeiros and Jomier (1993) e Medeiros and Jomier (1994), nos quais é introduzida o conceito de versionamento onde cada versão é um nó de um DAG e que haveria uma cadeia de ponteiros ligando esses nós, materializando o grafo e as possíveis evoluções dos dados, nas suas várias possibilidades. Muitas das abstrações apresentadas nos trabalhos não foram expressas em maiores detalhes, sendo estes artigos introdutórios no corpo de conhecimento de versionamento aplicado a dados geográficos. Durante a análise, foi percebido, em alguns momentos, que os modelos propostos tem seu contexto misturado com banco de dados geográfico temporais e análise temporal dos dados, fruto da época em que os artigos foram propostos. Este fato não é desejado, já que banco de dados temporais e versionamento de dados não se caracterizam com o mesmo escopo. Enquanto o primeiro foca no ordenamento cronológico da validade dos dados, o segundo visa a integração de trabalho cooperativo feito por mais de um usuário sobre a base.

Devido aos escassos trabalhos versando sobre versionamento aplicado a dados geográficos os trabalhos apresentados na presente Seção abordam tópicos muito heterogêneos de versionamento, como o uso de identificador espacial (SID) para correlação entre versões, complementação de modelagem para permitir múltiplas versões e algoritmo de *diff* para dados geográficos, não permitindo um comparativo adequado ente eles. Cabe ressaltar que nenhum dos trabalhos apresenta contribuições a respeito de manipulação da geometria quando a mesma é alterada nos dois *branches*.

3.2 Softwares Relacionados

Nesta seção são apresentados os *softwares* que foram desenvolvidos para tratar o problema de versionamento de dados espaciais. Alguns dos softwares aqui apresentados já não estão mais disponíveis, mas foram incluídos neste trabalho, pois estão referenciados na literatura e possuem características relevantes à proposta desta dissertação. Outros encontram-se disponíveis, contudo não há mais manutenção do código em seu repositório, sugerindo que o projeto está descontinuado. Outros estão em fases iniciais de desenvolvimento e, mesmo possuindo ferramentas funcionais, ainda não tem todas as

funcionalidades necessárias para uso em produção de dados vetoriais.

3.2.1 GitSpatial

A primeira ferramenta a ser abordada é o projeto **GitSpatial**², que esteve ativo entre 2013 e 2015 e, desde então, não é mais atualizado. O site referência do projeto³ também não se encontra mais ativo. Pelo exame da documentação ainda disponível, é possível constatar que o projeto pretendia usar o conjunto GIT + GitHub para fazer o versionamento de dados GeoJSON, sendo seu código desenvolvido principalmente em *Python*, porém não há muito mais detalhes. Apesar de não ser possível fazer testes na ferramenta para verificar detalhes de funcionamento de particularidades do sistema, é possível afirmar que tal ferramenta teria uma limitação substancial ao escopo pretendido, pois não trabalha com banco de dados, o que não é desejado para o presente trabalho.

3.2.2 GeoGig

A segunda ferramenta a ser abordada é o projeto **GeoGig**⁴, que iniciou em meados de 2011 ainda com o nome de GeoGit, sendo renomeado por volta de 2014 para o nome atual. O projeto se define como *Geospatial Distributed Version Control System* e, segundo seu site, é “uma ferramenta *open source* inspirada no GIT, mas adaptada para o contexto de trabalho com dados geográficos”. O projeto permite o versionamento, uso como *working copy*, não só de arquivos comuns a comunidade de dados geográficos, como *Shapefile* e *Spatialite*, mas também permite o versionamento de dados em banco de dados PostGIS. Além de permitir a WC em PostGIS, ele também possibilita a criação de repositório, local e/ou remoto, em estrutura de pastas/arquivos ou em banco de dados Postgres.

Desenvolvido em *Java*, o projeto de fato apresenta muita inspiração no GIT, com estruturas e funcionalidades como *Tree*, *Commit*, *Branch*, Repositório local/remoto, WC. Um fato notado durante seu uso foi o tipo de monitoramento da WC, que no sistema precisa da ação do usuário para indicar que deve ser feita uma varredura para identificar as mudanças. No GIT, esse monitoramento da WC é de modo ativo, no qual qualquer

²<<https://github.com/JasonSanford/gitspatial>>

³<<https://gitspatial.com/>>

⁴<<https://geogig.org>>

mudança nos arquivos contidos na WC já se encontra disponível para inclusão na *Staging Area*, diferente do que ocorre no GeoGig. Assim, todas as mudanças feitas na WC devem posteriormente passar por uma fase de inclusão na *Staging Area*, o que em termos de funcionalidade não representa necessariamente um problema. Em termos de velocidade, o GeoGig apresentou certa lentidão, pois quando se está trabalhando em base de dados ainda relativamente pequena (6 mil Km² - cerca de 35 cartas 1:25.000 do SCN) essas operações já apresentam um tempo de processamento de mais de 30 minutos. Cabe ressaltar que tal base de dados continha mais de 830 mil feições (220 mil pontos, 290 mil linhas e 320 mil áreas). Com o aumento de volume da base de dados, o sistema apresentou comportamento irregular, passando a gerar erros aleatórios de funcionamento, que não permitiram o prosseguimento dos testes em produção. Uma forma de mitigar os problemas seria reduzir o volume de dados a ser trabalhado pelo sistema com o uso das funcionalidades de *shallow clone* e *sparse clone*, porém os testes indicaram que tais ferramentas não funcionavam corretamente.

Outro ponto que foi possível observar é que o sistema estabelece como *Atomic Data Unit* os atributos da feição. Assim, em *branch* distintos, uma tupla só estaria em conflito caso um mesmo atributo fosse alterado de maneira distinta por ambos. Na hipótese de atributos diferentes de uma tupla serem unilateralmente alterados em cada *branch*, os mesmos não geram conflitos durante o *merge*. Essa escolha permite maior grau de integração automatizada, porém com menor grau de controle. Apesar de bastante completo em termos de funcionalidades, o lento ciclo de desenvolvimento do sistema tem sugerido que o projeto se encontra descontinuado.

3.2.3 Kart

A próxima ferramenta a ser comentada é o sistema **Kart**⁵, em desenvolvimento desde início de 2019 ainda com o nome de SNO e renomeado em meados do primeiro semestre de 2021 para Kart. O projeto se define como um *Distributed version-control* para dados geoespaciais e tabulares. O projeto permite o versionamento de banco PostGIS, além de MySQL, SQL Server e arquivos GeoPackage, que tem internamente um SpatiaLite para armazenar os dados, com a previsão de inclusão de mais formatos. Seu repositório é estruturado na configuração de pastas/arquivos, baseado na estrutura utilizada pelo GIT.

⁵<<https://kartproject.org/>>

Escrito principalmente na linguagem *Python*, o Kart é uma das ferramentas que tem desenvolvimento mais ativo no presente momento. Porém, ainda é um projeto que, embora esteja funcionando, não atende plenamente as necessidades do presente trabalho. Contudo, em seu *road map* já estão inseridas algumas funcionalidades que vão acrescentar importantes funções ao projeto, como a possibilidade de *partial clone*. Seu uso com WC em PostGIS apresentou um desempenho surpreendente. Em testes feitos com os dados citados (830 mil feições) um *import* do banco completo com a criação de WC PostGIS foi concluída em cerca de 5 minutos e a efetivação de um *commit* foi concluída em segundos. Quanto às escolhas de implementação, foram observados alguns pontos que podem explicar o desempenho obtido. Por exemplo, o uso de *trigger* para monitoramento das mudanças ocorridas no banco após um *commit*, armazenando em uma tabela auxiliar as informações de quais tabelas/id mudaram. Esse tipo de escolha cria um monitoramento ativo do banco e agiliza a criação de um novo *commit*. Outra escolha observada foi a opção por não estabelecer um *Staging Area*, o que pode acarretar em perda de flexibilidade na hora de definir quais alterações fazem parte de um determinado *commit*. É possível afirmar ainda que a escolha do sistema pela criação de WC interna ao sistema, pelo menos no presente estado de desenvolvimento, tem implicado em perdas de modelagem na construção da WC que podem afetar a funcionalidade do modelo replicado na WC. Ainda, como última observação, foi identificado que o sistema estabelece como *Atomic Data Unit* toda a tupla, o que permite um maior controle da informação a ser integrada, mas pode, potencialmente, gerar maior quantidade de conflitos por ocasião do *merge*.

3.2.4 GeoDiff + Mergin

A última solução a ser apresentada é o conjunto formado pela biblioteca **GeoDiff**⁶ e a plataforma **Mergin**⁷, aplicações desenvolvidas para fazer a função de um sistema que permite versionar e integrar dados coletados em campo. O GeoDiff é uma biblioteca construída para lidar com *diff* de dados geoespaciais, cuja linguagem principal é *Python* e que teve seu início em julho de 2020. Já o Mergin é uma plataforma que o usuário armazena e sincroniza dados geoespaciais, sendo desenvolvida em *Javascript* e *Python*, e que só em junho de 2021 tornou seu código público. A biblioteca GeoDiff é usada para manipular os dados, possibilitando criar *changesets* (*diff*) entre dois *dataset* e, também, aplicar, inver-

⁶<<https://github.com/lutraconsulting/geodiff/>>

⁷<<https://public.cloudmergin.com/>>

ter, concatenar *changesets*, além de outras funções que lidam diretamente com os dados. Já a plataforma atua armazenando os estados e *changeset* entre duas versões dos dados, o que permite o trabalho colaborativo entre mais de um usuário, possibilitando o controle por usuário além de algumas outras funções de gerência.

O uso somente da biblioteca GeoDiff, por seu modo de funcionamento de comparação entre dois arquivos/banco de dados, apresenta alguns inconvenientes, como por exemplo, lidar somente com bases de dados materializadas. Dessa forma, é necessário ter duas versões completas do arquivo ou duas instâncias do banco para usar a biblioteca: uma com a versão atual/corrente, na qual foram feitas as mudanças durante o trabalho; outra com a versão que é com a qual deve ser feita a comparação, geralmente a última versão salva. Outro problema seria a forma de comparação, o modo de definir o *diff*, neste caso, o chamado *2 way*. Quando é feita a comparação de modo sequencial, em uma base de dados que evolui de modo linear e incremental, esse modo funciona sem maiores problemas. Porém, quando a estrutura de trabalho estabelece que uma base de dados pode apresentar evoluções em paralelo, a semelhança de um *branch*, o modo *2 way* não funciona bem, pois desconsidera o estado anterior ao início da ramificação, o que torna o gerenciamento desses estados paralelos muito mais custoso. Os sistemas atuais de DVCS trabalham com comparações de modo *3 way*, em que comparam os dois estados atuais e o estado anterior comum, definindo melhor o que de fato é conflito e o que é evolução unilateral de uma das bases, possibilitando o *merge* de modo mais automático. Vale lembrar que parte das limitações da biblioteca são supridas pela plataforma Mergin, que permite fazer às vezes do DVCS, porém um pouco mais limitado. Um exemplo dessa limitação ocorre no momento de lidar com estados paralelos, no qual o sistema força o usuário a atualizar primeiro toda sua estrutura, priorizando as atualizações disponíveis na plataforma para as bases que ele está trabalhando e, somente depois, acomodando suas próprias atualizações, para finalmente conseguir enviar o novo arquivo de *diff* para a plataforma.

3.2.5 Análise Comparativa

Para promover um melhor entendimento das soluções apresentadas, as mesmas são comparadas e depois analisadas com base na proposta da presente dissertação. As soluções de versionamento de dados são comparados com relação ao tipo de WC que pode ser usada, tipo de repositório e forma de comparação para integração dos dados geográficos. Começando com o *GitSpatial*, o mesmo só permitia o uso de GeoJSON como

Tabela 3.1: Comparativo entre sistemas de versionamento de dados geográficos disponíveis

	GitSpatial	GeoGig	Kart	GeoDiff + Mergin
WC	GeoJSON	Shapefile SpatialLite PostGIS	PostGIS MySQL SQL Server GeoPackage	* GeoPackage PostGIS
Repositório	estrutura de pastas/arquivos	estrutura de pastas/arquivos Postgres	estrutura de pastas/arquivos	**
Tipo de comparação	igualdade estrita	igualdade estrita	igualdade estrita	resolução manual

* - Não pode ser considerada uma WC como os outros sistemas, mas permite o uso com os formatos citados.

** - O *Mergin* atua como repositório dos dados, porém não há como estabelecer paralelo com os outros sistemas.

da WC. Pelo fato de usar o GIT como sistema de versionamento, é possível afirmar que seu repositório é em estrutura de arquivos, podendo ser remota (com o uso do GitHub), e ainda que o tipo de comparação para integração dos dados é feita por igualdade estrita. O *GeoGig* permite o uso de WC em formato Shapefile, SpatialLite e PostGIS. Permite ainda o uso de repositório tipo estrutura de arquivos ou Postgres, e o tipo de comparação para integração também é feita por igualdade estrita. O *Kart* permite o uso de PostGIS, MySQL, SQL Server e GeoPackage como WC, porém só permite o uso de estrutura de pastas para o repositório. Já a solução *GeoDiff + Mergin* tem um funcionamento um pouco diferente dos sistemas anteriores. O *GeoDiff* é uma biblioteca na qual é possível trabalhar dados em GeoPackage e PostGIS, e o *Mergin* funciona como repositório das versões. O tipo de integração dos dados também difere dos outros sistemas, pois a carga de trabalho de integração é passada para o usuário.

Com base nos pontos elencados é possível verificar que tanto o *GeoGig* quanto o *Kart* apresentam as soluções mais elaboradas. Porém, ainda que tenham diversos fatores favoráveis, por fazerem uso de comparação por igualdade estrita, mesmo para o atributo geometria, as mesmas não trazem refinamentos no trato da componente geográfica do dado. Assim a proposta apresentada nesta dissertação difere das soluções apresentadas e permite que se favoreça o trato da componente geográfica, permitindo a integração de dados que, para os presentes sistemas, são admitidos como conflito.

4 DEFINIÇÕES PRELIMINARES

Este capítulo apresenta de forma sistematizada conceitos importantes relacionados ao tema desta dissertação, introduzindo as operações de GIS possíveis de serem executadas nas geometrias. O Capítulo apresenta também o comportamento atual de comparação e integração de geometria encontrado em sistemas de versionamento durante o *merge*, mesmo nos sistemas especializados em dados geográficos. A proposta desta dissertação estende comportamento atual dos sistemas de versionamento de dados. Tanto a sistematização das operações de GIS como a definição do comportamento de comparação e integração de geometria nos sistemas atuais fazem parte das contribuições da presente dissertação.

4.1 Operação de GIS sobre geometrias

Para a melhor compreensão da presente dissertação são apresentadas as principais formas de interação com os dados vetoriais, com base nas operações de GIS existentes. De modo a possibilitar a melhor compreensão das interações, as mesmas são demonstradas visualmente tendo como base um banco de dados espacial, no qual as geometrias são armazenadas, o qual segue a modelagem do mapeamento sistemático brasileiro, que representa rios, rodovias, entre outros tipos de objetos que são previstos na modelagem. Para tanto, algumas convenções são elencadas:

- uma base de dados é estruturada em camadas/classes, que têm atributos, sendo essa estrutura um esquema definido pela modelagem a ser usada no mapeamento. No caso do mapeamento sistemático brasileiro, essa modelagem é definida pela *Especificação Técnica para Estruturação de Dados Geoespaciais Vetoriais* (ET-EDGV) (DIRETORIA DE SERVIÇO GEOGRÁFICO, 2017), que no momento se encontra em sua terceira versão;
- as camadas/classes são nomeadas de acordo com o tipo de objeto armazenado, porém, em geral, os objetos são agregados em classes mais genéricas. Por exemplo, na modelagem brasileira são previstas as classes “Trecho_Rodoviario”, “Trecho_Drenagem”, “Vegetacao_Cultivada”, “Edificacoes”, entre outros;
- cada classe tem conjuntos distintos de atributos. Neste capítulo, adotamos como identificador para cada classe o atributo “id”. Existe também o atributo “geom”

para armazenar a geometria, que podem ser do tipo ponto, linha ou área, definido para a classe como um todo. Esses atributos armazenam informações dos objetos a serem inseridos na base de dados. Como exemplo, podemos dizer que a classe “Trecho_Drenagem” tem os atributos “id”, “geom”, “nome”, “regime” (hidrológico) e “salinidade”. Já a classe “Trecho_Rodoviario” tem os atributos “id”, “geom”, “nome”, “tipo” (tipo de estrada), “jurisdicao”, entre outros;

- feição é a forma de nomear um objeto de mundo real que sendo inserido na base de dados se materializa como um registro / tupla no banco de dados. A feição agrega a geometria e os atributos numéricos e textuais referentes ao objeto de mundo;
- supondo que existe uma base de dados inicial (Situação Inicial), incompleta, de uma região, a partir dela são feitas as alterações/operações que leva a base de dados para estados intermediários seguintes (Situação Intermediária 1, Situação Intermediária 2, . . . , Situação Intermediária N) de construção da base de dados. Ao final, a base de dados se encontra em sua Situação Final, que contempla todas as operações feitas sobre os dados.

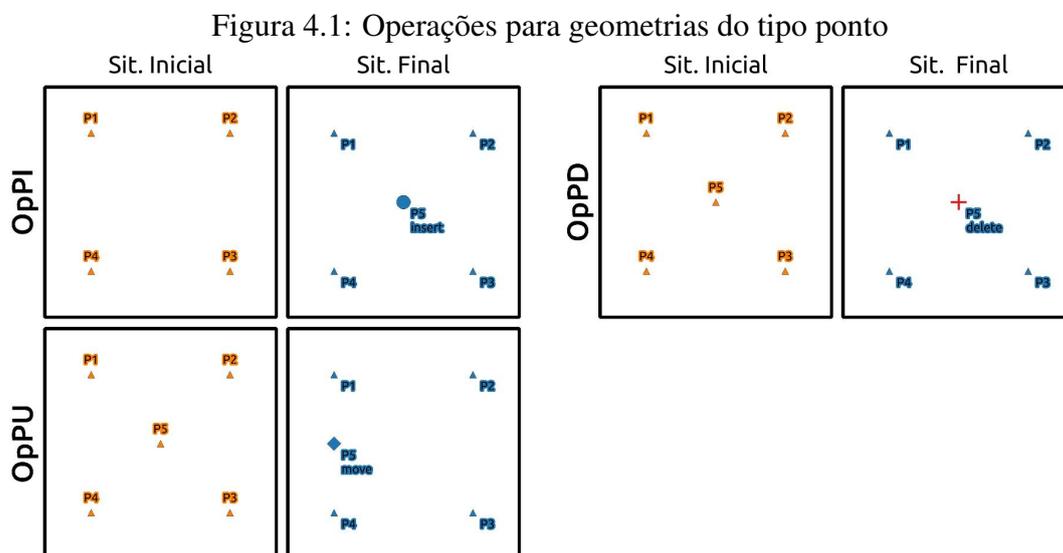
Apesar de previsto em diversas modelagens de dados para mapeamento, a exemplo da ET-EDGV do Brasil, como o foco da proposta desta dissertação recai sobre as alterações que envolvem a geometria, para as definições seguintes, a modelagem é simplificada de modo que a camada só tenha o atributo identificador e a geometria, sendo também referente a uma classe genérica, sem uma temática específica.

Com base na convenção apresentada, a seguir, são estabelecidas as possíveis operações a serem executadas em base de dados espaciais, com referência a cada tipo geométrico. As figuras que ilustram as subseções a seguir apresentam a esquerda a situação inicial dos dados e a direita a situação final, após executada a operação que é indicada. As figuras servem para demonstrar visualmente qual o comportamento de cada uma das operações definidas.

4.1.1 Operações para camadas do tipo ponto

Considerando os pontos $id = P1$, $id = P2$, . . . , $id = Pn$, todos com posições independentes uns dos outros. Com base em uma classe de geometria ponto, as possíveis operações de GIS a serem executadas são:

1. alteração no registro de feições:
 - (a) *insert* de feição (OpPI). O *insert* de uma feição é a adição de um registro na classe;
 - (b) *delete* de feição (OpPD). O *delete* de uma feição é a remoção de um registro da classe;
2. alterações na geometria:
 - (a) *Update* de feição por movimentação do ponto (OpPU). Alteração em que o ponto passa a ter novos valores de coordenadas.



A Figura 4.1 apresenta a visualização das operações descritas anteriormente. É possível expressar todas as alterações na camada por meio das três operações: *insert* de feição, *delete* de feição, *update* por movimentação do ponto. Assim, de posse de dois estados de uma camada de geometria ponto de um banco de dados espacial, é possível descrever todas as alterações na geometria por um conjunto dessas operações.

4.1.2 Operações para camadas do tipo linha

Considerando as linhas $id = L_1, id = L_2, \dots, id = L_n$, todas independentes, cada linha L_n é composta pelos vértices $V_0 L_n, V_1 L_n, \dots, V_k L_n$, também independentes entre si. A indicação $V_0 L_n$ é uma referência ao vértice de índice 0 da linha cujo identificador da linha é L_n , já que não há um identificador específico dos vértices que compõem a linha.

Assim o índice do vértice na composição da linha é usado para referenciar os vértices da linha. Com base em uma classe de geometria linha, as possíveis operações de GIS a serem executadas são:

1. alteração no registro de feições:
 - (a) *insert* de feição (OpLI). O *insert* de uma feição é a adição de um registro na classe, a semelhança do OpPI.
 - (b) *delete* de feição (OpLD). O *delete* de uma feição é a remoção de um registro na classe, a semelhança do OpPD.

2. alterações que envolvem *update* da geometria, por segmentos:
 - (a) *update* de feição por *insert* de vértice no início/fim (OpLUIV1) ou segmento intermediário da linha (OpLUIV2). O *update* por *insert* de vértice é a adição de um vértice em uma geometria que já existe, aumentando o número de vértices que compõem a geometria.
 - (b) *update* de feição por *delete* de vértice no início/fim (OpLUDV1) ou segmento intermediário da linha (OpLUDV2). O *update* por *delete* de vértice é a remoção de um vértice em uma geometria que já existe, diminuindo o número de vértices que compõem a geometria.

3. alterações que envolvem *update* da geometria, de modo global:
 - (a) *update* de feição por *flip* (OpLUF). A operação de *flip* é uma operação na qual ocorre a inversão do ordenamento completo dos vértices. Para entender esta operação é necessário primeiro entender que a linha representa um objeto físico no mundo real, e a depender do objeto sua direção tem importância. Tomando como exemplo uma camada que representa os rios de uma determinada região, cuja geometria é do tipo linha. Naturalmente, os rios tem um sentido de deslocamento, de montante para jusante. Assim, pode ser interessante que a geometria que representa o rio também tenha direção, para representar essa característica do objeto, o que pode favorecer outros tipo de interação como determinar a bacia de contribuição de um determinado ramo de um rio. Para que seja possível alterar a direção de uma geometria do tipo linha existe essa operação de GIS de inversão do ordenamento total dos vértices.

Figura 4.2: Operações básicas para geometrias do tipo linha

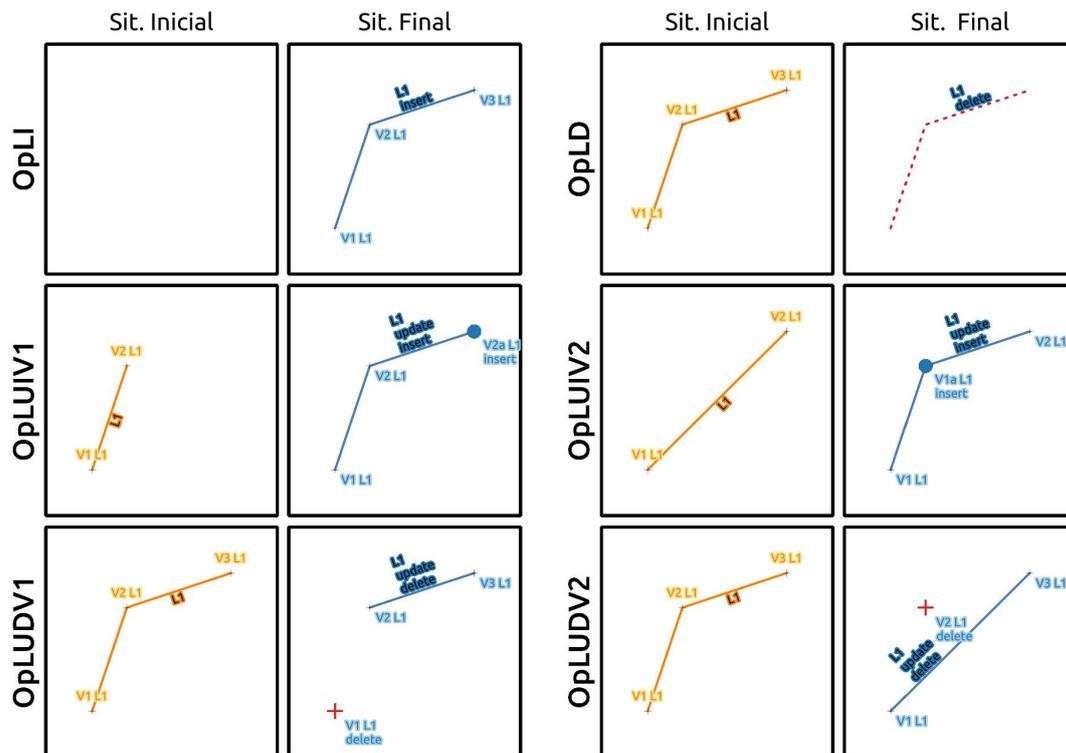
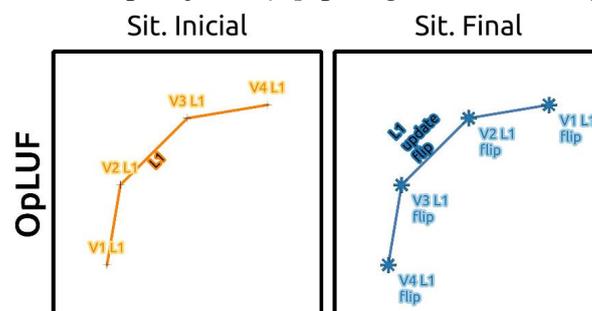


Figura 4.3: Operação de *flip* para geometrias do tipo linha



4. alterações que envolvem *update* da geometria, por operações compostas (que podem ser descritas por sequência de operações mais simples):

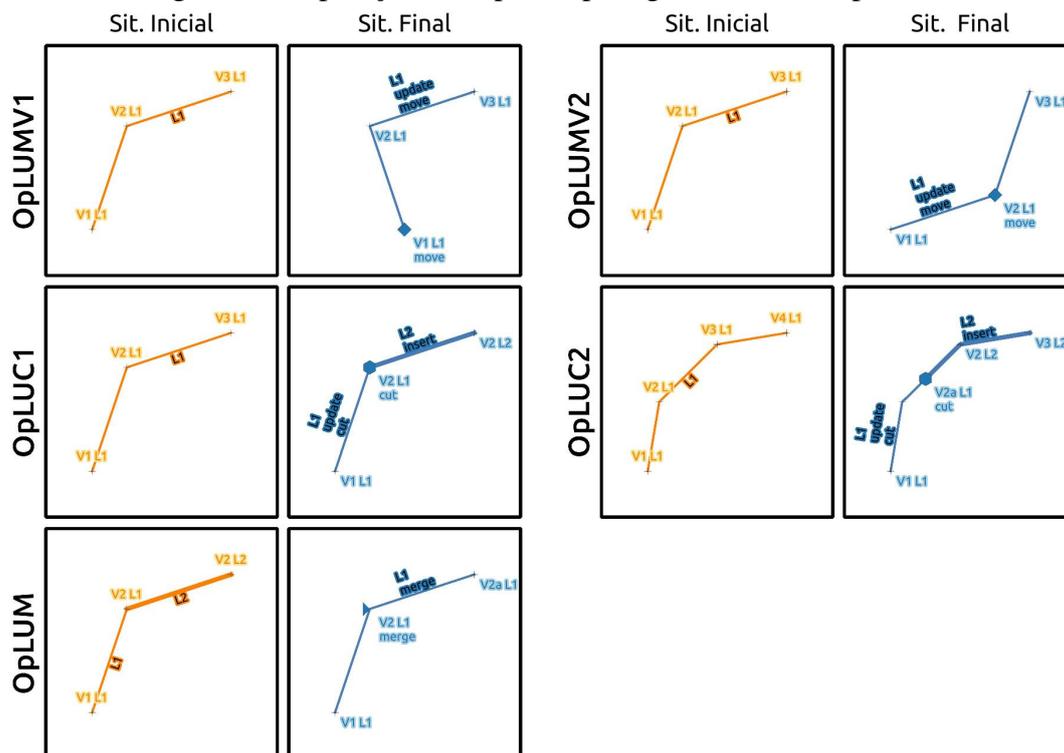
- (a) *update* de feição por movimentação de vértice no início/fim (OpLUMV1) ou vértice intermediário da linha (OpLUMV2). A operação de movimentação de vértice pode ser expressa pela composição de duas operações mais simples, uma que é o *delete* de vértices e a outra que é o posterior *insert* de vértices nas posições desejadas, mantendo a mesma quantidade e sequência.
- (b) *update* de feição por *cut* no vértice (OpLUC1) ou no segmento (OpLUC2). A operação de *cut* em vértice é composta por duas operações mais simples, uma que é o *delete* de vértices da feição L1, e o *insert* da feição L2. Já a operação de *cut* em segmento é composta por três operações mais simples: o *delete* dos vértices em L1, depois o *insert* de vértices intermediário na feição L1, e o *insert* da feição L2.
- (c) *update* de feição por *merge* (OpUM). A operação de *merge* é composta por duas operações mais simples, uma que é o *delete* da feição L2 e o *insert* de vértices oriundos de L2 na feição L1.

As Figuras 4.2, 4.3, e 4.4 ilustram as operações apresentadas. Diante das diversas operações especificadas para geometrias do tipo linha, é possível expressar todas as alterações em uma camada por meio de cinco operações: *insert* de feição, *delete* de feição, *update* por *insert* de vértice, e *update* por *delete* de vértice, além da operação de *update* de feição por *flip*. Assim, de posse de dois estados de uma camada de geometria linha de um banco de dados espacial, é possível descrever todas as alterações por um conjunto dessas operações. Cabe ressaltar que nesta Seção são definidos os conceitos das operações de linha, porém a demonstração das operações, que é o core desta dissertação é apresentado no Capítulo 5.

4.1.3 Operações para camadas do tipo área

Considerando as áreas $id = A_1, id = A_2, \dots, id = A_n$, todas independentes. Cada área A_n é composta obrigatoriamente por uma linha fechada externa (*outer ring*) $L_0 A_n$,

Figura 4.4: Operações compostas para geometrias do tipo linha



e por zero, uma ou muitas linhas fechadas internas (*inner rings*) $L_1 An, L_2 An, \dots, L_v An$, independentes entre si, porém sem haver sobreposição ou toque entre elas e também internas ao polígono definido por $L_0 An$, conforme preconizado na norma OGC - *Simple Feature Access* (OGC, 2011). Cada linha fechada interna define um *hole* no polígono, sendo os *holes* nomeados de acordo com sua linha de definição, $H_1 An, H_2 An, \dots, H_v An$, respectivamente. Os *holes* são regiões onde a área ali representada não se verifica no terreno. Por exemplo, ao construir um rio em que sua geometria seja uma área, as ilhas que possam existir ao longo do rio formam *holes* na geometria que representa o rio. A ordem da numeração das linhas internas é definida pela sequência presente na definição da área em WKT. Exemplo:

- Polígono sem *hole* = POLYGON ((30 60, 40 90, 20 90, 10 70, 30 60))
 - Linha externa (L_0)= (30 60, 40 90, 20 90, 10 70, 30 60)
- Polígono com 1 *hole* = POLYGON ((60 60, 70 90, 50 90, 40 70, 60 60),(50 70, 60 80, 60 70, 50 70))
 - Linha externa (L_0)= (60 60, 70 90, 50 90, 40 70, 60 60)
 - Linha interna 1 (L_1)= (50 70, 60 80, 60 70, 50 70)

- Polígono com mais de 1 *hole* = POLYGON ((60 60, 70 90, 50 90, 40 70, 60 60),(50 70, 60 80, 60 70, 50 70),(50 80, 50 85, 60 85, 50 80))
 - Linha externa (L_0)= (60 60, 70 90, 50 90, 40 70, 60 60)
 - Linha interna 1 (L_1)= (50 70, 60 80, 60 70, 50 70)
 - Linha interna 2 (L_2)= (50 80, 50 85, 60 85, 50 80)

Cada linha L_v An (externa ou interna) é composta pelos vértices $V_1 L_v$ An, $V_2 L_v$ An, ..., $V_k L_v$ An, independentes entre si, para cada L_v An. Com base em uma classe de geometria área, as possíveis operações de GIS a serem executadas são:

1. alteração no registro de feições:

- (a) *insert* de feição (OpAI). O *insert* de uma feição é a adição de um registro na classe, a semelhança do OpPI / OpLI.
- (b) *delete* de feição (OpAD). O *delete* de uma feição é a remoção de um registro na classe, a semelhança do OpPD / OpLD.

2. alterações que envolvem *update* da geometria, por alteração na quantidade de *hole*:

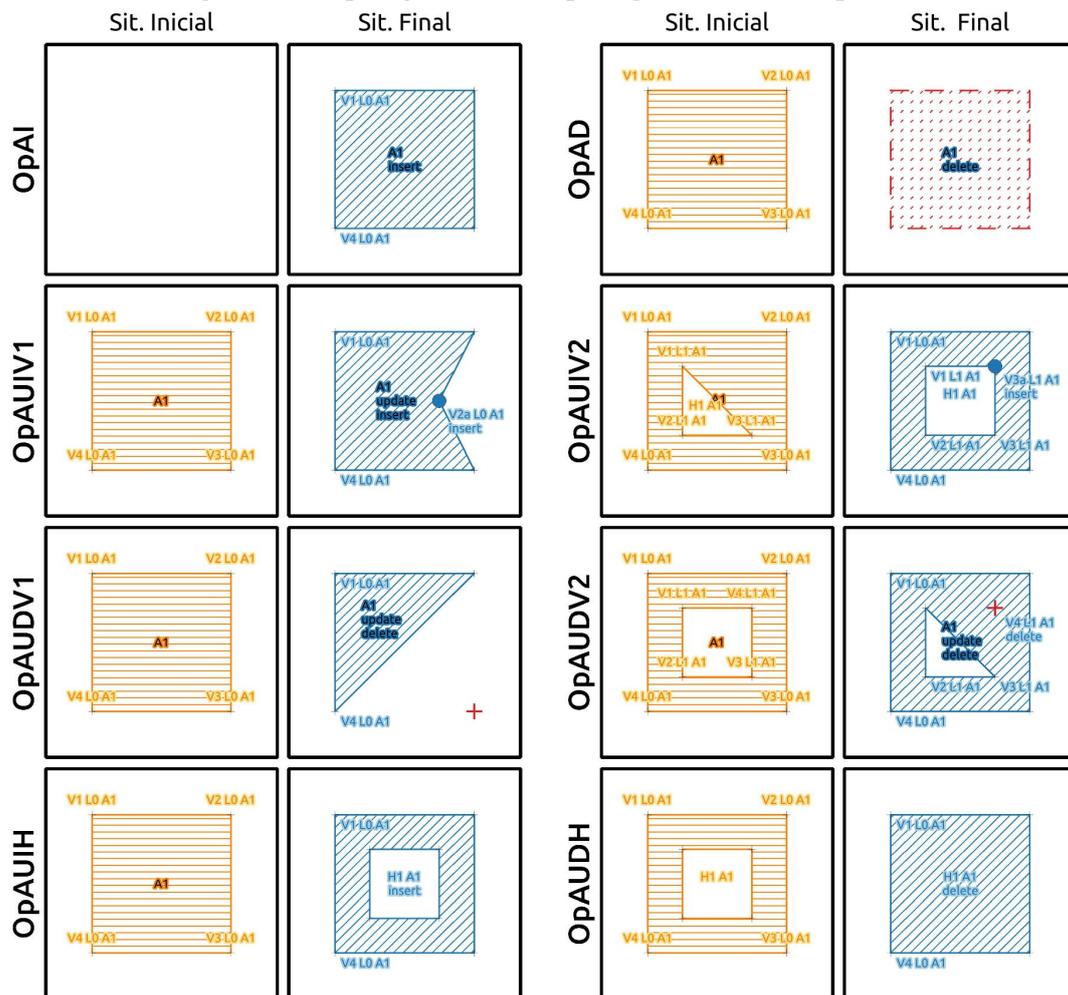
- (a) *update* de feição por *insert* de *hole* (OpAUIH). Esta operação promove a criação de um *hole* em uma área.
- (b) *update* de feição por *delete* de *hole* (OpAUDH). Esta operação permite que se remova um *hole* de um local onde se verifica que o mesmo não é mais necessário.

3. alterações que envolvem *update* da geometria, por alteração nos segmentos das linha fechadas:

- (a) *update* de feição por *insert* de vértice, no *outer ring* (OpAUIV1) ou *inner ring* (OpAUIV2), a semelhança do OpLUI;
- (b) *update* de feição por *delete* de vértice, no *outer ring* (OpAUDV1) ou *inner ring* (OpAUDV2), a semelhança do OpLUD;

4. alterações que envolvem *update* da geometria, por operações compostas:

Figura 4.5: Operações básicas para geometrias do tipo área

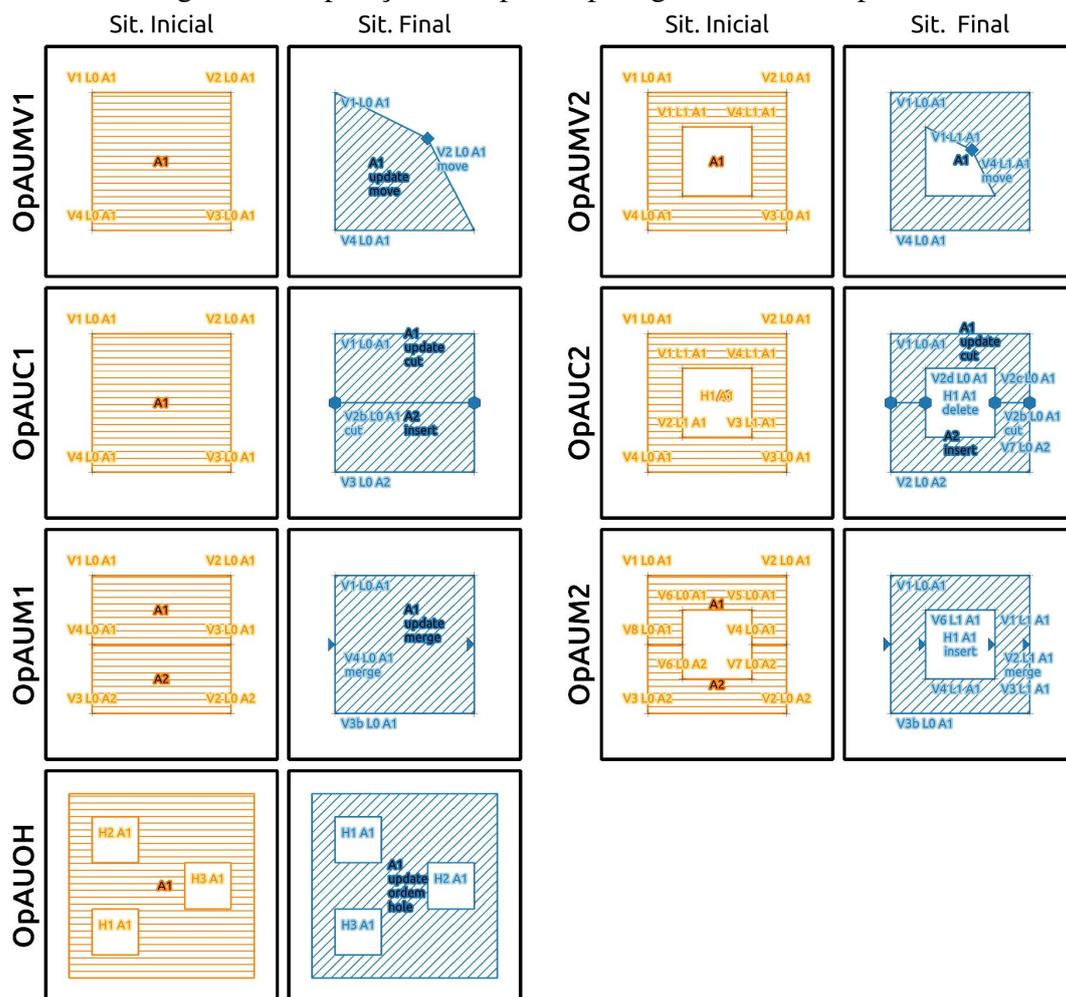


- (a) *update* de feição por movimentação de vértice, no *outer ring* (OpAUMV1) ou *inner ring* (OpAUMV2), a semelhança do OpLUP. A operação de movimentação de vértice, quando for expressa pela composição de duas operações mais simples, uma que é o *delete* de vértices e a outra que é o posterior *insert* de vértices nas posições desejadas.
- (b) *update* de feição por *cut*, podendo envolver somente o *outer ring* (OpAUC1) ou também o *outer ring* e *inner ring* (OpAUC2), a semelhança da OpLC. A operação de *cut* quando envolve somente o *outer ring* (OpAUC1) é composta por uma sequência de operações mais simples, uma que é o *delete* podendo ser seguido de *insert* de vértices na feição A1 e o *insert* da feição A2. Já a operação de *cut* quando envolve também o *inner ring* (OpAUC2) é composta por uma sequência de operações mais simples: *delete* do *hole* envolvido, *delete* de vértices de A1, *insert* de vértices em A1 (obrigatório para moldar a região que seria o remanescente do *hole*), e o *insert* da feição A2.
- (c) *update* de feição por *merge*, podendo envolver somente o *outer ring* (OpAUM1) ou também geometrias que formem um *inner ring* (OpAUM2), a semelhança da OpLM. A operação de *merge* quando envolve somente o *outer ring* (OpAUM1) é composta por uma sequência de operações mais simples, uma que é o *delete* da feição A2 e o *delete* e *insert* de vértices na feição A1. Já a operação de *merge* quando envolve geometrias que venha a formar um *inner ring* (OpAUM2) é composta por uma sequência de operações mais simples: *delete* da feição A2, *delete* e *insert* de vértices em A1 e o *insert* do *hole* em A1.
- (d) *update* de feição por alteração de ordem dos *holes* (OpAUOH). Esse tipo de alteração ocorre quando há mudança na ordem de definição dos *holes* entre as versões da área. Em geral, essa alteração é ocasionada de forma indireta, porém a operação de alteração de ordem dos *holes* pode ser retratada como composta por uma ou várias operações *delete* de *hole* seguidas por *insert* de *hole*, alterando assim a ordem em que os mesmos são definidos na geometria.

5. alteração indireta da geometria:

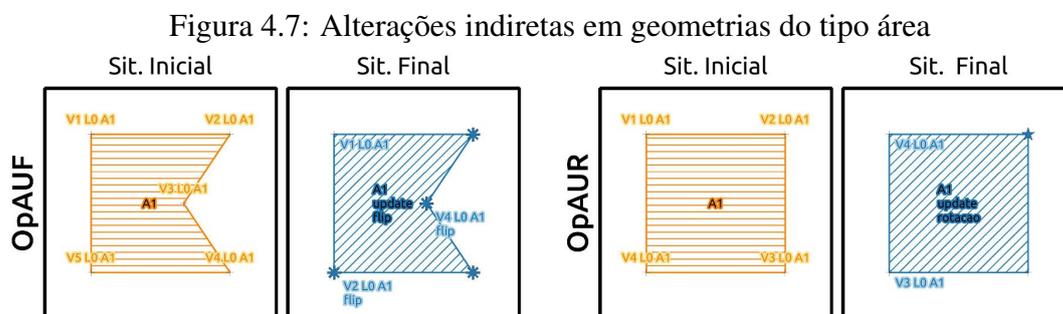
Todas as alterações anteriores são frutos de possíveis operações de GIS aplicadas na geometria diretamente pelo operador, ainda que algumas delas possam ocorrer

Figura 4.6: Operações compostas para geometrias do tipo área



indiretamente quando são executadas sequências de processamentos na geometria. A seguir, serão apresentadas alterações que ocorrem somente de maneira não intencional, pela combinação de operações.

- (a) *update* de feição por *flip* do *ring* (OpAUF). O *flip* para áreas não possui uma operação de GIS específica e ocorre pela combinação de outras operações. Não há sentido físico na direção do *ring* de área, já que são sempre linhas fechadas delimitando uma região.
- (b) *update* de feição por alteração de rotação de índice de *ring* (OpAUR). A rotação de índice de um *ring* também não possui uma operação de GIS específica e ocorre pela combinação de outras operações. Um ponto a considerar é que uma geometria do tipo área delimita uma região do espaço, e portanto a linha que a delimita não tem necessariamente um ponto físico de início/fim, porém a sua implementação computacional necessita desse artifício de ordenamento dos pontos, com a consequência de ter um ponto inicial/final.



Ainda que não exista uma operação de GIS específica para as alterações indiretas, é necessário especificá-las para que seja possível estabelecer a relação entre dois estados de uma base de dados que contém áreas. Caso não fossem especificadas, uma simples alteração do tipo *flip*, que ocorra em uma geometria, seria impossível de ser tratado pela proposta desta dissertação.

As Figuras 4.5, 4.6 e 4.7 ilustram as operações apresentadas. Diante das diversas operações especificadas para geometrias do tipo áreas, é possível expressar todas as alterações em uma camada área por meio de oito operações: *insert* de feição, *delete* de feição, *update* por *insert* de *hole*, e *update* por *delete* de *hole*, *update* por *insert* de vértice, e *update* por *delete* de vértice, além de duas alterações que podem ocorrer de maneira

indireta, *update* de feição por *flip* e *update* de feição por rotação de índice de *ring*. Assim, de posse de dois estados de uma camada de geometria área de um banco de dados, é possível descrever todas as alterações por um conjunto dessas operações. Cabe ressaltar que nesta Seção são definidos os conceitos das operações de áreas, porém demonstração das operações, que é o core desta dissertação é apresentado no Capítulo 5.

4.2 Dados Geográficos em Sistemas de Versionamento de Dados

Uma vez definida as operações de GIS possíveis sobre os dados, o próximo conjunto de definições preliminares para apoio ao entendimento do presente trabalho são as situações que ocorrem quando os dados são utilizados em um sistema de versionamento. Importante destacar que embora existam dois tipos de comportamento em relação ao *Atomic Data Unit* nos sistemas disponíveis de versionamento de dados geográficos, nenhum deles consegue integrar as geometrias se as mesmas não forem estritamente iguais (em sua definição WKT, ou equivalente). O primeiro é o comportamento que estabelece a tupla do banco de dados como unidade atômica e o segundo é o comportamento que estabelece o atributo da tupla como unidade atômica. Embora haja diferenças práticas entre os dois comportamentos, no tocante às mudanças complementares de geometrias, nenhum dos dois casos consegue resolver a integração das geometrias. Isso se deve ao fato que, no fim, a comparação entre as geometrias é feita pela simples comparação do texto de definição da geometria (por exemplo, o WKT), mesmo nos sistemas que trabalham com dados geográficos. Essa comparação do tipo estrita é um comportamento que gera impacto e não favorece o trabalho com dados geográficos. A partir dessa visão, são analisadas as interações dos dados em sistema de versionamento. Para esta parte do trabalho, são elencadas as seguintes convenções:

- as explicações apresentadas se referem a operação de *merge* do sistema de versionamento, no momento de integrar os dados dos dois *branches*. Assim, para entendimento completo são analisados quatro estados: o estado das feições no último ponto comum aos dois *branches* a serem trabalhados (*Branch Ancestor* - (B^A)); o estado das feições com as alterações feita pelo *operador 1* em seu próprio *branch* - (B^1) ; o estado das feições com as alterações feita pelo *operador 2* em seu próprio *branch* - (B^2) ; e o último estado que representa a união dos *branches* B^1 e B^2 (*Branch Merged* - (B^M)) ou pelo menos sua tentativa.
- para focar melhor nos possíveis problemas referentes as geometrias que podem ocorrer durante o processo de *merge* do versionamento é admitido que *branches* distintos não geram identificadores que possam entrar em conflito. Caso tal proposição não fosse admitida, poderiam ocorrer conflitos por uso de mesmo identificador para feições criadas em *branches* distintos, o que não é desejável. Para que tal problema não ocorra, podem ser adotadas diferentes estratégias como chave primária para os

dados, inclusive o uso de UUID (*universally unique identifier*), identificador que para fins práticos pode ser considerado único entre diferentes base de dados.

Com base nas convenções anteriores serão estabelecidas algumas das possíveis interações no momento do *merge* do versionamento, com referência a cada tipo geométrico. As interações apresentadas são exclusivamente focadas nos casos em que os usuários realizaram o mesmo tipo de operação nas versões. Cabe ressaltar que o foco desta dissertação é na interação das operações realizadas pelos usuários. Alterações que sejam distintas resultam em conflitos a serem resolvidos e devem ser assim tratadas, pois há intenção divergente por parte de cada um dos usuários. As figuras que ilustram as subseções a seguir apresentam no primeiro quadro a esquerda a situação dos dados no último ponto comum entre os *branches*; no segundo quadro da esquerda para a direita, a situação dos dados no *branch* do usuário 1; no terceiro quadro da esquerda para a direita, a situação dos dados no *branch* do usuário 2; no último quadro, à direita, a situação dos dados na situação de integração dos dados ou sua tentativa, o *branch merged*. As figuras servem para demonstrar visualmente qual o comportamento de cada um dos casos do ponto de vista do versionamento em sistemas atuais.

4.2.1 Versionamento para camadas de geometrias do tipo ponto

Nesta subseção são apresentados os possíveis casos de versionamento de dados cuja geometria é do tipo ponto. São apresentados os casos onde são feitas alterações nos dados por meio de operações de GIS, sendo apresentadas as operações que promovam alterações no registro, bem como alterações por meio de operação de *update*. As Figuras 4.8, 4.9 e 4.10 ilustram as diferentes interações de versionamento para as geometrias do tipo ponto.

1. Possíveis casos de alteração no registro das feições:

(a) Alteração por *insert* de feição:

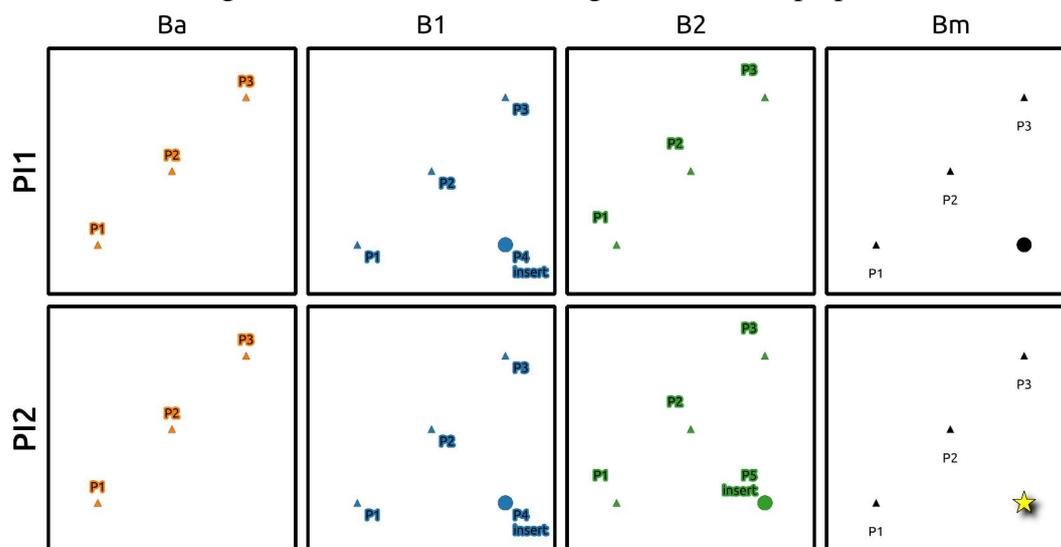
- i. se B^1 insere um ponto P4 e B^2 não altera a base de dados, não há conflito e a situação final é a estabelecida por B^1 (caso PI1). O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 insere um ponto P4 e B^2 também insere um ponto P4 em posições

diferentes, representando objetos diferentes do mundo real, existe o conflito por uso da mesma chave primária. Porém, como estabelecido no início da presente seção, o sistema de versionamento não permite que *branches* distintos gerem identificadores que possam gerar conflito. Dessa forma, a situação apresentada não se concretiza, e com o uso de identificador distinto o problema recai sobre dois casos como o apresentado pelo caso PI1, um em cada *branch*;

- iii. se B^1 insere um ponto P4 e B^2 insere um ponto P5, na mesma posição, tem-se um caso bem particular para dados geográficos. Se os dois elementos, cada um inserido em um *branch*, se referem a um mesmo elemento no mundo a ser mapeado, as duas feições na verdade são uma só e, portanto, há uma redundância de informação. Porém, não há como detectar esse tipo de conflito durante a integração/*merge* da base de dados, já que os identificadores são diferentes. Este é um caso em que é necessário executar um pós-processamento das informações para remoção de redundância. Dessa forma, a situação final é a estabelecida pela composição de B^1 e B^2 , com a necessidade do pós-processamento de deduplicação (caso PI2);
- iv. se B^1 insere um ponto P4 e B^2 também insere um ponto P4, na mesma posição, não haveria um conflito. Porém, como estabelecido no início da presente seção, o sistema de versionamento não permite que *branches* distintos gerem identificadores que possam gerar conflito. Dessa forma, a situação apresentada não se concretiza e, com o uso de identificador distinto, o problema recai sobre o caso anterior, caso PI2.

(b) Alteração por *delete* de feição:

- i. se B^1 deleta P2, e B^2 não altera P2, não há conflito e a situação final é a estabelecida por B^1 (caso PD1). O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 deleta P2 e B^2 também deleta P2, não há conflito, e a situação final é a estabelecida por B^1/B^2 (caso PD2);
- iii. se B^1 deleta P2 e B^2 deleta P3, não há conflito, e a situação final é a estabelecida pela composição de B^1 e B^2 (caso PD3).

Figura 4.8: Casos de *insert* de geometrias do tipo ponto

2. Possíveis casos de *update*:

(a) Alteração de movimentação de posição da geometria:

- i. se B^1 altera a posição de P2, e B^2 não altera P2, não há conflito e a posição final de P2 é a estabelecida por B^1 (caso PU1). O mesmo raciocínio se aplicaria caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a alteração fosse feita somente por B^2 ;
- ii. se B^1 altera posição de P2 e B^2 também altera a posição de P2 para a mesma posição, não há conflito, e a posição final de P2 é a nova posição estabelecida por B^1/B^2 (caso PU2);
- iii. se B^1 altera a posição de P2 e B^2 também altera a posição de P2, porém para outra posição, neste caso, tem-se um conflito que precisa ser resolvido no *merge* do versionamento, já que cada *branch* executou uma alteração na posição de P2 para uma posição distinta (caso PU3).

4.2.2 Versionamento para camadas do tipo linha

Nesta subseção são apresentados os possíveis casos de versionamento de dados cuja geometria é do tipo linha. São apresentados os casos onde são feitas alterações nos dados por meio de operações de GIS, sendo apresentadas as operações que promovam

Figura 4.9: Casos de *delete* de geometrias do tipo ponto

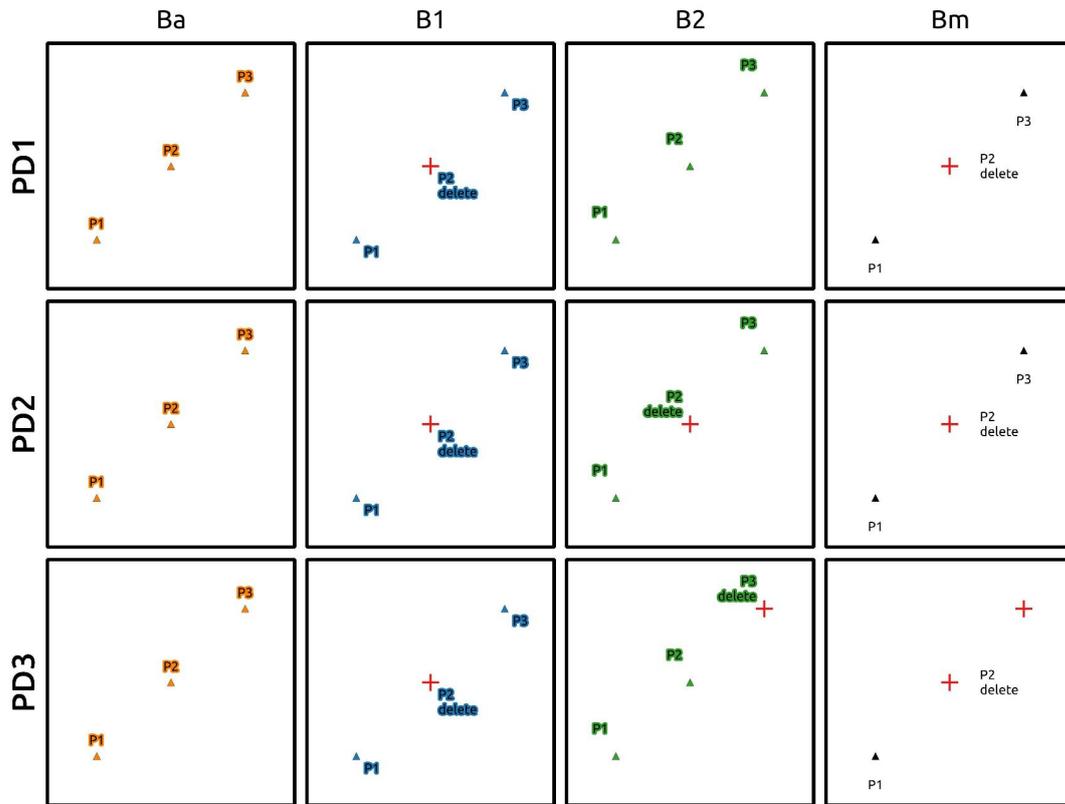
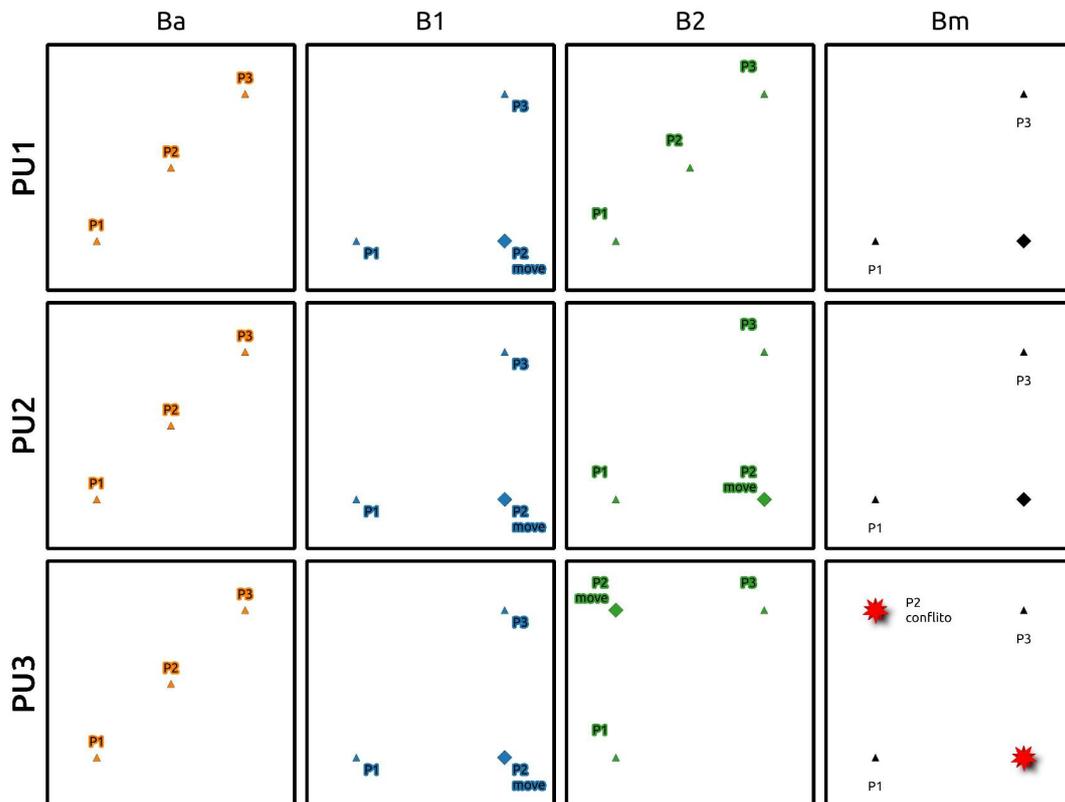


Figura 4.10: Casos de *update* de geometrias do tipo ponto



alterações no registro, bem como alterações por meio de operação de *update*. As Figuras 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17 e 4.18 ilustram as diferentes interações de versionamento para as geometrias do tipo linha.

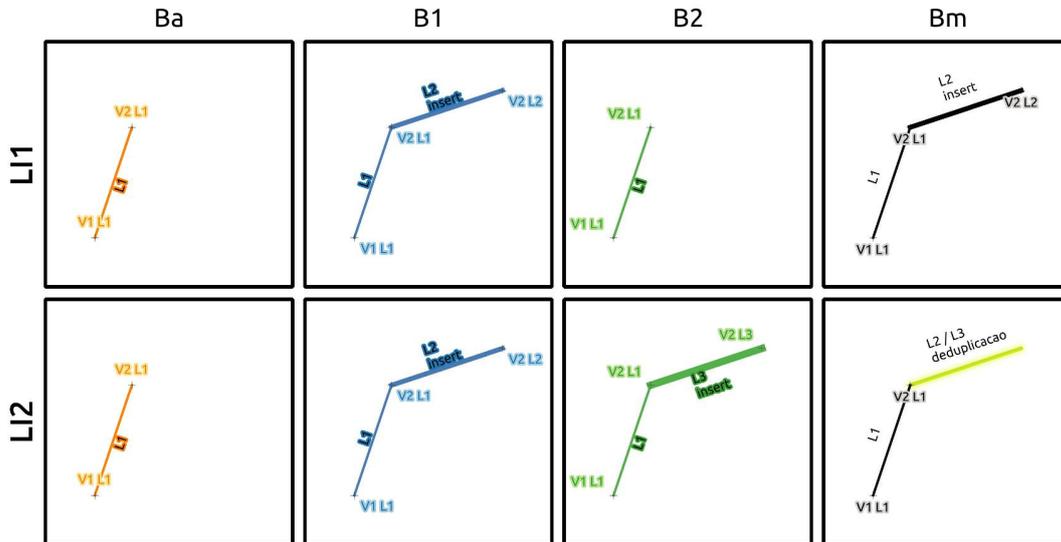
1. Possíveis casos de alteração no registro das feições:

(a) Alteração por *insert* de feição:

- i. se B^1 insere uma linha L2, e B^2 não altera a base de dados, não há conflito e a situação final é a estabelecida por B^1 (caso LI1). O mesmo raciocínio se aplicaria caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 insere uma linha L2 e B^2 também insere uma linha L2 em posição diferente, representando objetos diferentes do mundo real, existe o conflito por uso da mesma chave primária. Porém, como estabelecido no início da presente seção, o sistema de versionamento não permite que *branches* distintos gerem identificadores que possam gerar conflito. Dessa forma, a situação apresentada não se concretiza e, com o uso de identificador distinto, o problema recai sobre dois casos como o apresentado pelo caso LI1, um em cada *branch*;
- iii. se B^1 insere uma linha L2 e B^2 insere uma linha L3, na mesma posição, tem-se um caso bem particular para dados geográficos. Se os dois elementos, cada um inserido em sua base, se referem a um mesmo elemento no mundo real a ser mapeado, as duas feições na verdade são uma só e, portanto, há uma redundância de informação. Porém, não há como detectar esse tipo de conflito durante a integração/*merge* da base de dados, já que os identificadores são diferentes. Este é um caso em que é necessário executar um pós-processamento das informações para remoção de redundância. Dessa forma, a situação final é a estabelecida pela composição de B^1 e B^2 , com a necessidade do pós-processamento de deduplicação de dados (caso LI2). Deduplicação de dados é uma técnica para busca e remoção de dados em redundância em uma base de dados. Esta atividade não é foco da presente dissertação;
- iv. se B^1 insere uma linha L2 e B^2 também insere uma linha L2, na mesma posição, não há um conflito. Porém, como estabelecido no início da presente seção, o sistema de versionamento não permite que *branches* dis-

tintos gerem identificadores que possam gerar conflito. Dessa forma, a situação apresentada não se concretiza e, com o uso de identificador distinto, o problema recai sobre o caso anterior, caso LI2.

Figura 4.11: Casos de *insert* de geometrias do tipo linha



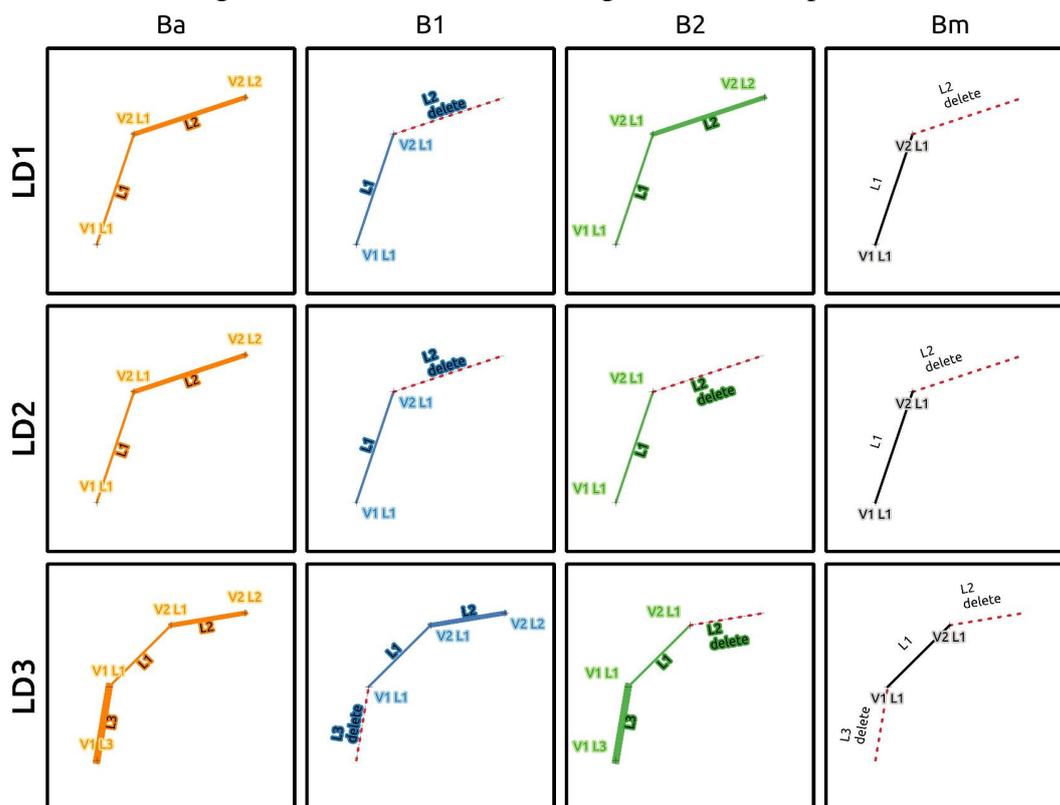
(b) Alteração por *delete* de feição:

- i. se B^1 deleta L2, e B^2 não altera L2, não há conflito e a situação final é a estabelecida por B^1 (caso LD1). O mesmo raciocínio se aplicaria caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 deleta L2 e B^2 também deleta L2, não há conflito, e a situação final é a estabelecida por B^1/B^2 (caso LD2);
- iii. se B^1 deleta L2 e B^2 deleta L3, não há conflito, e a situação final é a estabelecida pela composição de B^1 e B^2 (caso LD3).

2. Possíveis casos de *update*:

(a) Alteração por *insert* de vértices:

- i. se B^1 insere vértices em L1, e B^2 não altera L1, não há conflito e a geometria final de L1 é a estabelecida por B^1 . Esse tipo de caso é possível ocorrer tanto no início/fim das linhas (caso LUIV1a) como no meio das linhas (caso LUIV1b). Ressaltando que o mesmo raciocínio se aplica caso

Figura 4.12: Casos de *delete* de geometrias do tipo linha

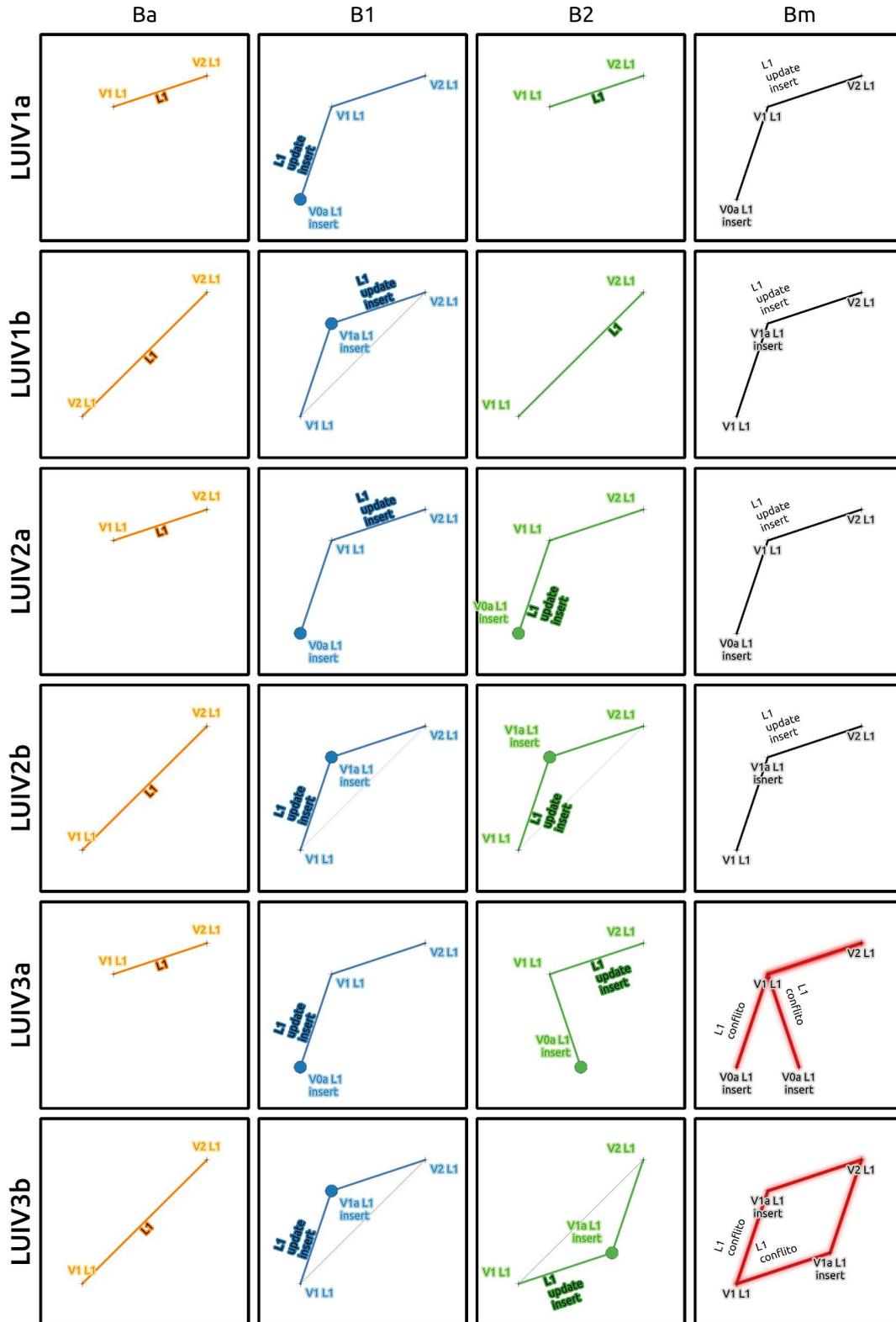
fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;

- ii. se B^1 insere vértices em L1 e B^2 também insere vértices em L1 da mesma forma, não há conflito, e a geometria final de L1 é a nova geometria estabelecida por B^1/B^2 . Esse tipo de caso é possível ocorrer tanto no início/fim das linhas (caso LUIV2a) como no meio das linhas (caso LUIV2b);
- iii. se B^1 insere vértices em L1, e B^2 também insere vértices em L1, porém de modo diferente, neste caso, tem-se um conflito que precisa ser resolvido no *merge* do versionamento, já que cada *branch* executou uma alteração distinta. Esse tipo de caso é possível ocorrer tanto no início/fim das linhas (caso LUIV3a) como no meio das linhas (caso LUIV3b).

(b) Alteração por *delete* de vértices:

- i. se B^1 deleta vértices em L1, e B^2 não altera L1, não há conflito e a geometria final de L1 é a estabelecida por B^1 . Esse tipo de caso é possível ocorrer tanto no início/fim das linhas (caso LUDV1a) como no meio das

Figura 4.13: Casos de *update* por *insert* de vértices em geometrias do tipo linha



linhas (caso LUDV1b). O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;

- ii. se B^1 deleta vértices de L1 e B^2 também deleta vértices em L1 da mesma forma, não há conflito, e a geometria final de L1 é a nova geometria estabelecida por B^1/B^2 . Esse tipo de caso é possível ocorrer tanto no início/fim das linhas (caso LUDV2a) como no meio das linhas (caso LUDV2b).
- iii. se B^1 deleta vértices em L1 e B^2 também deleta vértices de L1, porém de modo distinto, neste caso, tem-se um conflito que precisa ser resolvido no *merge* do versionamento, já que cada *branch* executou uma alteração distinta. Esse tipo de caso é possível ocorrer tanto no início/fim das linhas (caso LUDV3a) como no meio das linhas (caso LUDV3b).

(c) Alteração por *flip*:

- i. se B^1 promove uma alteração do tipo *flip* em L1, e B^2 não altera L1, não há conflito e a geometria final de L1 e L2 é a estabelecida por B^1 (caso LUF1). O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 promove uma alteração do tipo *flip* em L1, e B^2 também altera L1 da mesma forma, não há conflito, e a geometria final de L1 é a nova geometria estabelecida por B^1/B^2 (caso LUF2).

(d) Alteração por movimentação de posição de vértices:

- i. se B^1 altera a posição de vértices de L1 e B^2 não altera L1, não há conflito e a geometria final de L1 é a estabelecida por B^1 (caso LUMV1). O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 altera a posição de vértices de L1 e B^2 também altera L1 da mesma forma, não há conflito, e a geometria final de L1 é a nova geometria estabelecida por B^1/B^2 (caso LUMV2);
- iii. se B^1 altera a posição de vértices de L1 e B^2 também altera a posição de vértices de L1, porém de modo diferente, neste caso, tem-se um con-

Figura 4.14: Casos de *update* por *delete* de vértice em geometrias do tipo linha

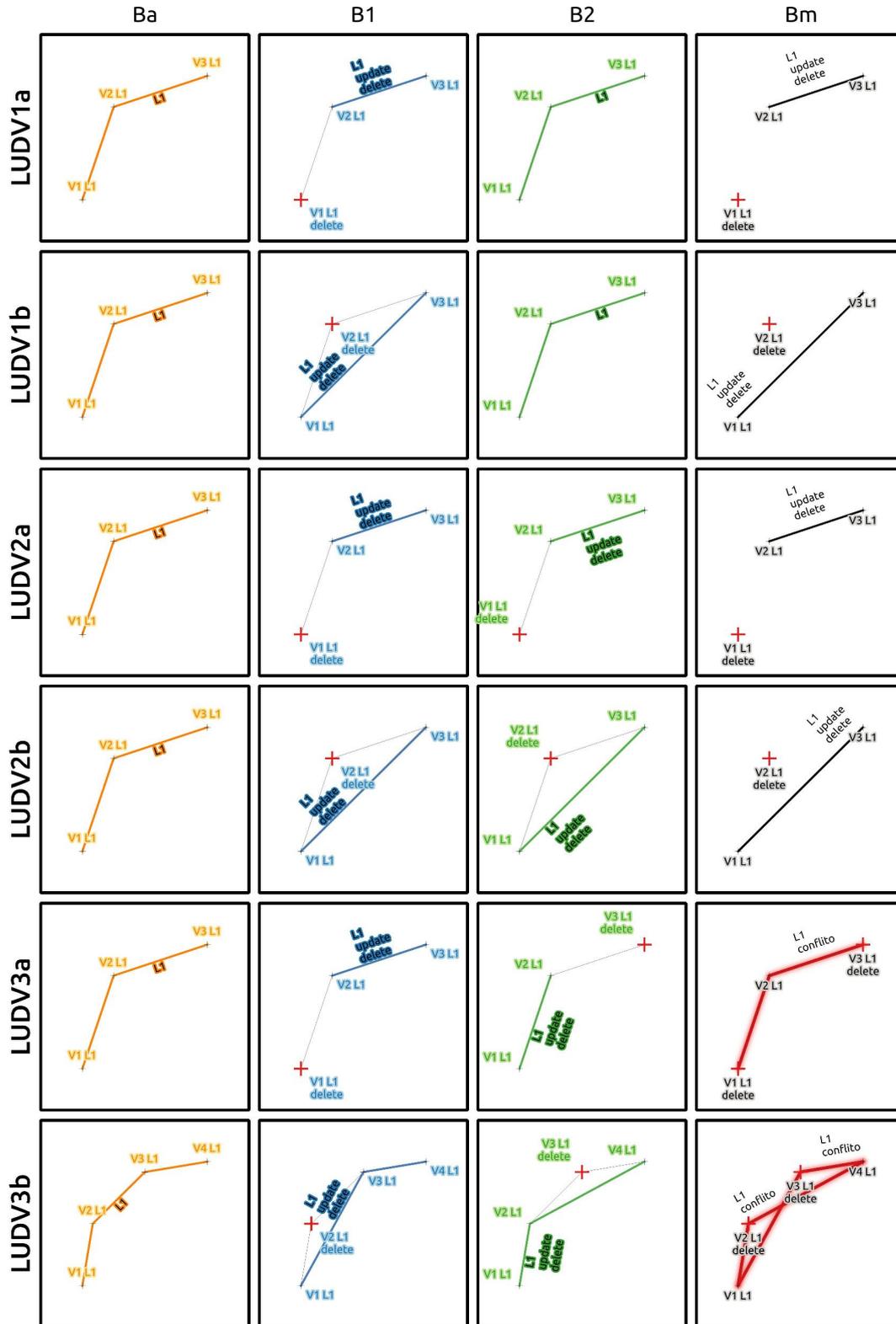
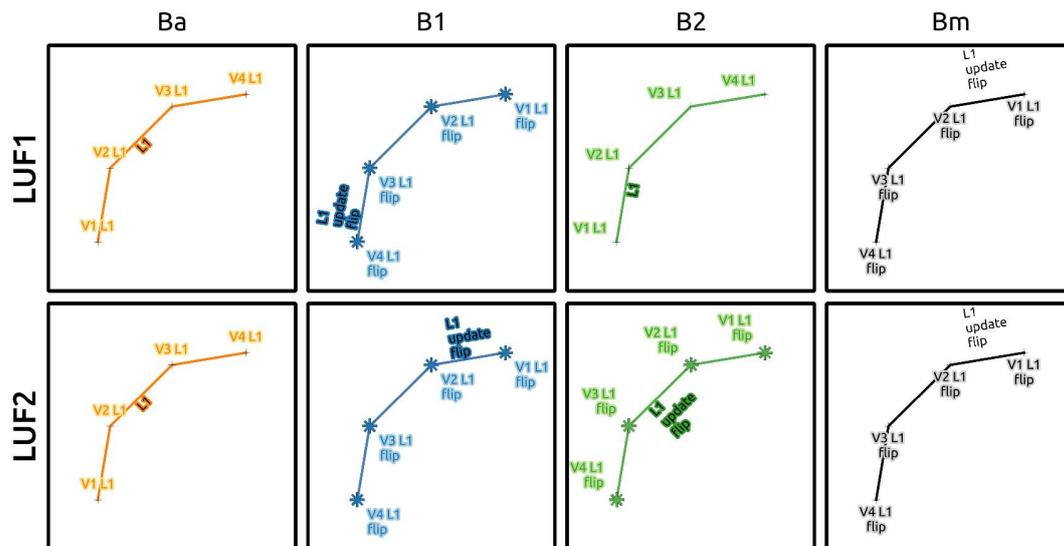
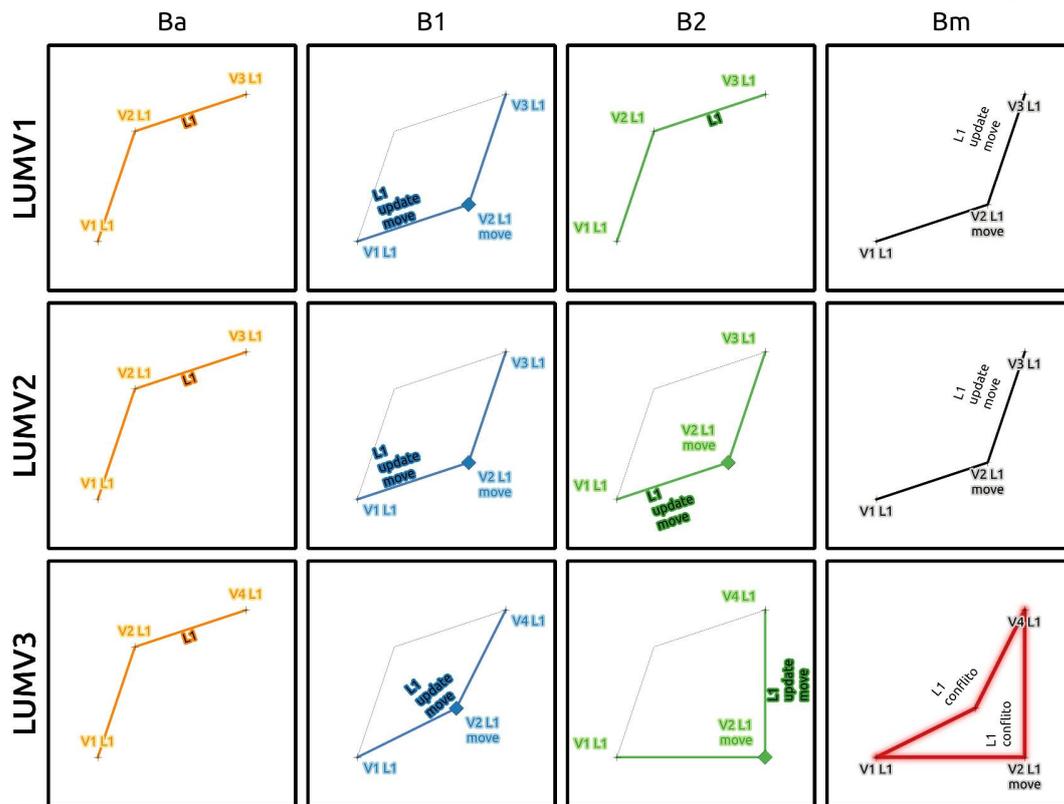


Figura 4.15: Casos de *update* por *flip* de geometrias do tipo linha

flito que precisa ser resolvido no *merge* do versionamento, já que cada *branch* executou uma alteração nos vértices de L1 de modo distinto (caso LUMV3).

(e) Alteração por *cut*:

- i. se B^1 corta L1 e B^2 não altera L1, não há conflito e a geometria final de L1 e L2 é a estabelecida por B^1 . Esse tipo de caso é possível ocorrer tanto por corte coincidente com alguns dos vértices (caso LUC1a) como por corte no meio dos segmentos (caso LUC1b). O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 corta L1 e B^2 também corta L1, na mesma posição, pode ocorrer ou não um conflito. Ao cortar uma geometria, uma das partes recebe o id original e a outra recebe um id novo. Assim, pode acontecer do mesmo id ser estabelecido para pedaços diferentes da geometria, o que causa um conflito. A seguir, são apresentados os dois casos: um quando o mesmo pedaço recebe o mesmo id (caso LUC2a), dessa forma não causa conflito, porém gera um caso de inserção de feição duplicada da outra parte a semelhança do caso LI2; e outro quando ocorre “inversão” de id e, por consequência, um conflito (caso LUC2b), e também novos casos LI2, pela inserção de feições duplicadas. Vale ressaltar que esses casos podem ocorrer tanto com o corte coincidente com algum dos vértices

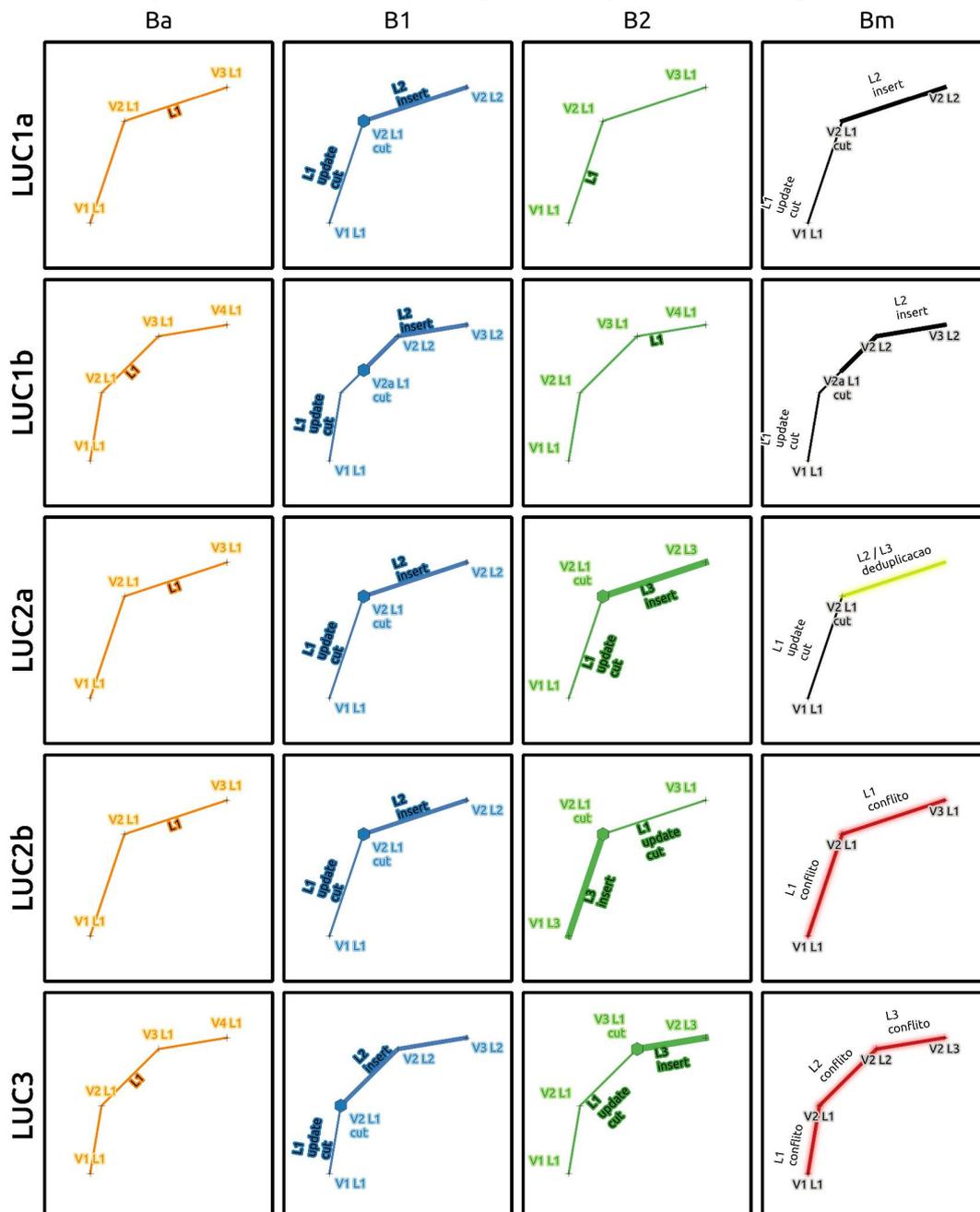
Figura 4.16: Casos de *update* por movimentação de vértice em geometrias do tipo linha

(como apresentado na Figura 4.17 - LUC2a e LUC2b) como quando o corte ocorre entre segmentos;

- iii. se B^1 corta $L1$, e B^2 também corta $L1$, porém de modo diferente, ocorre conflito, pois há alteração em $L1$ de modo diferente em cada *branch*. Esse tipo de caso é possível ocorrer tanto por corte coincidente com algum dos vértices como por corte no meio dos segmentos (caso LUC3).

(f) Alteração por operação de GIS do tipo *merge*:

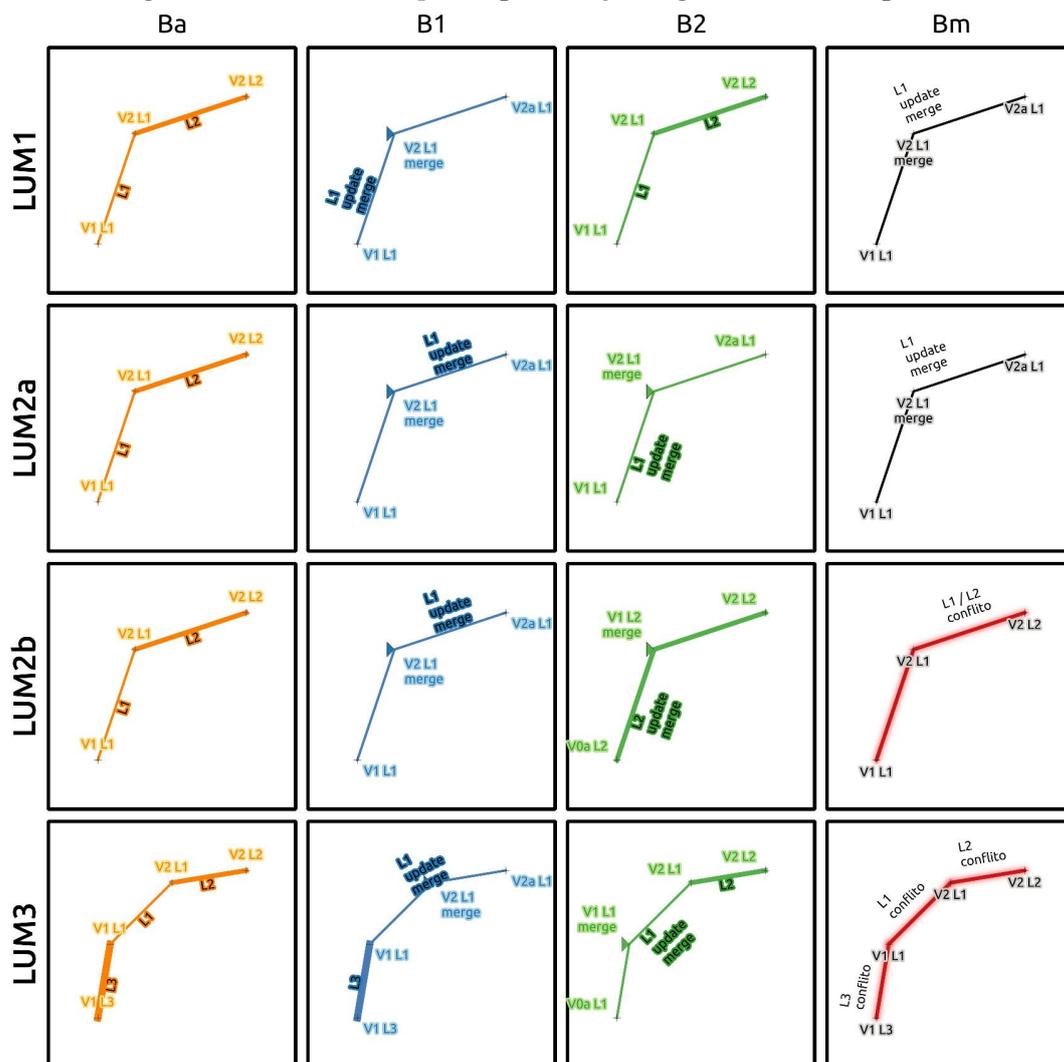
- i. se B^1 faz uma operação de GIS do tipo *merge* de $L1$ com $L2$, e B^2 não altera $L1$ nem $L2$, não há conflito e a geometria final de $L1/L2$ é a estabelecida por B^1 (caso LUM1). O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 faz uma operação de GIS do tipo *merge* de $L1$ com $L2$, e B^2 também faz uma operação de GIS do tipo *merge* de $L1$ com $L2$, pode ocorrer ou não um conflito. Ao fazer uma operação de GIS do tipo *merge* de geometrias, a geometria final recebe o id de uma das partes. Assim, pode

Figura 4.17: Casos de *update por cut* de geometrias do tipo linha

acontecer o caso da geometria final receber id oriundo de partes diferentes, o que causa conflito de versionamento. A seguir, são apresentados os dois casos: um quando não ocorre “inversão” de id (caso LUM2a) e outro quando ocorre a “inversão” de id e, por conseguinte, um conflito de versionamento (caso LUM2b).

- iii. se B^1 faz uma operação de GIS do tipo *merge* de L1 com L2, e B^2 também faz uma operação de GIS do tipo *merge*, porém de L1 com L3, vai ocorrer conflito, pois há alteração em L1 de modo diferente em cada *branch* (caso LUM3).

Figura 4.18: Casos de *update* por *merge* de geometrias do tipo linha



4.2.3 Versionamento para camadas de geometrias do tipo área

Nesta subsecção são apresentados os possíveis casos de versionamento de dados cuja geometria é do tipo área. São apresentados os casos onde são feitas alterações nos dados por meio de operações de GIS, sendo apresentadas as operações que promovam alterações no registro, bem como alterações por meio de operação de *update*. As Figuras 4.19, 4.20, 4.21, 4.22, 4.23, 4.24, 4.25, 4.26, 4.27, 4.28, 4.29 e 4.30 ilustram as diferentes interações de versionamento para as geometrias do tipo área.

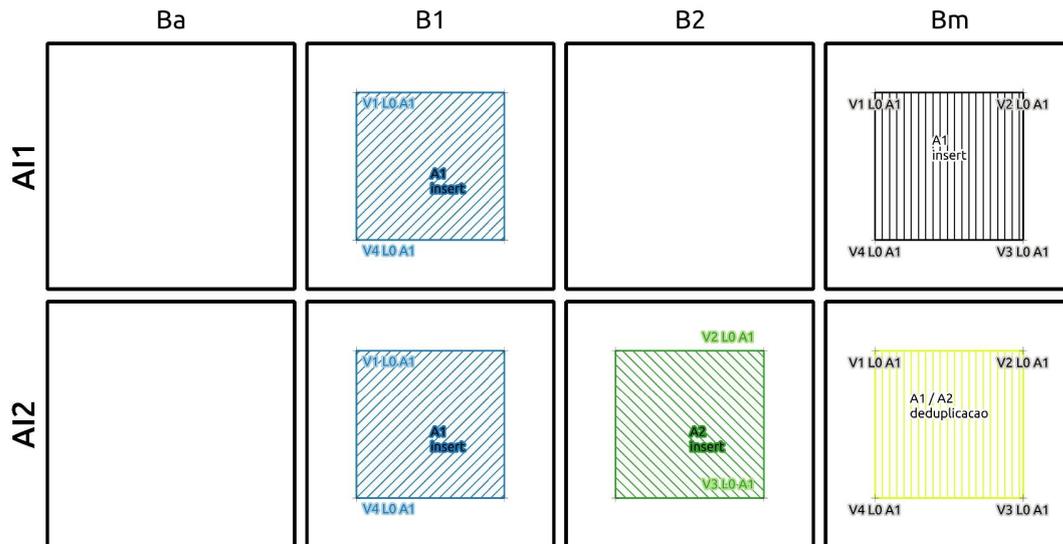
1. Possíveis casos de alteração no registro da feição:

(a) Alteração por *insert* de feição:

- i. se B^1 insere uma área A2 e B^2 não altera a base de dados, não há conflito e a situação final é a estabelecida por B^1 (caso AI1). O mesmo raciocínio se aplicaria caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 insere uma área A2 e B^2 também insere uma área A2 em posição diferente, representando objetos diferentes do mundo real, existe o conflito por uso da mesma chave primária. Porém, como estabelecido no início da presente seção, o sistema de versionamento não permite que *branches* distintos gerem identificadores que possam gerar conflito. Dessa forma, a situação apresentada não se concretiza e, com o uso de identificador distinto, o problema recai sobre dois casos como o apresentado pelo caso LI1, um em cada *branch*;
- iii. se B^1 insere uma área A2 e B^2 insere uma área A3, na mesma posição. Se os dois elementos, cada um inserido em sua base, se referem a um mesmo elemento no mundo real a ser mapeado, as duas feições na verdade são uma só e portanto uma redundância de informação. Porém, não há como detectar esse tipo de conflito durante a integração/*merge* da base de dados, já que os identificadores são diferentes. Este é um caso em que é necessário executar um pós-processamento das informações para remoção de redundância. Dessa forma, a situação final é a estabelecida pela composição de B^1 e B^2 , com a necessidade do pós-processamento de deduplicação (caso AI2)
- iv. se B^1 insere uma área A2 e B^2 insere uma área A2, na mesma posição,

não há conflito. Porém, como estabelecido no início da presente seção, o sistema de versionamento não permite que *branches* distintos gerem identificadores que possam gerar conflito. Dessa forma, a situação apresentada não se concretiza e, com o uso de identificador distinto, o problema recai sobre o caso anterior, caso LI2.

Figura 4.19: Casos de *insert* de geometrias do tipo área



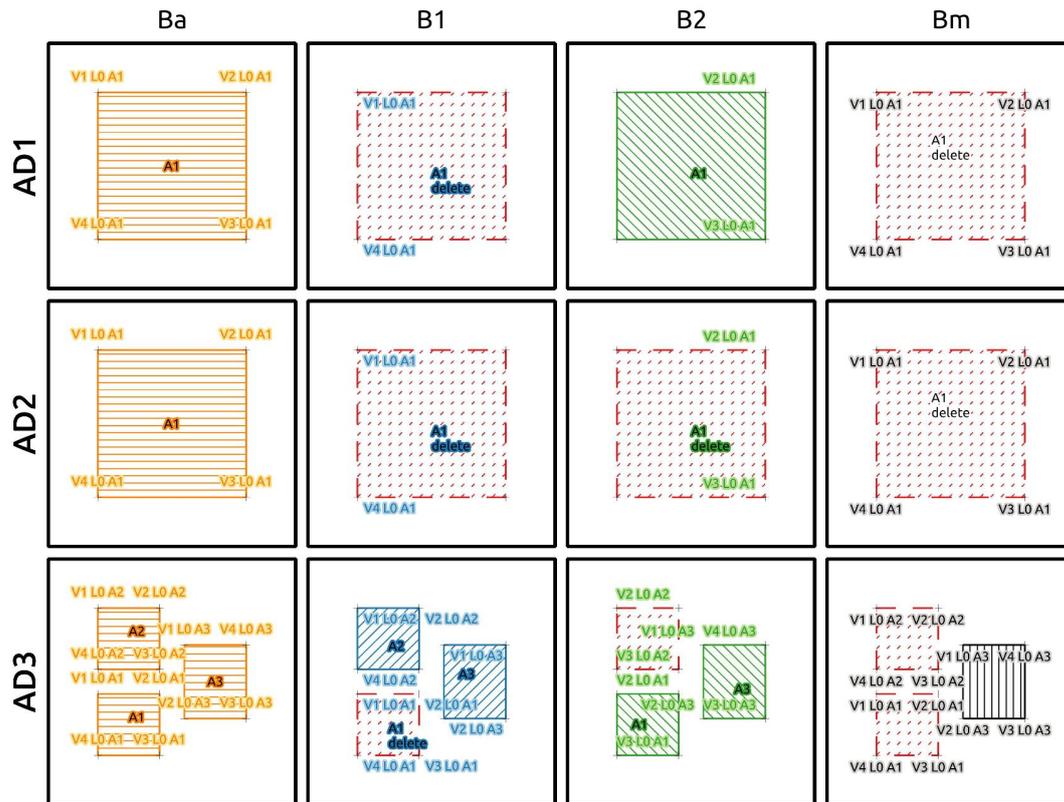
(b) Alteração por *delete* de feição:

- i. se B^1 deleta A2 e B^2 não altera A2, não há conflito e a situação final é a estabelecida por B^1 (caso AD1). O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 deleta A2 e B^2 também deleta A2, não há conflito e a situação final é a estabelecida por B^1/B^2 (caso AD2);
- iii. se B^1 deleta A2 e B^2 deleta A3, não há conflito e a situação final é a estabelecida pela composição de B^1 e B^2 (caso AD3).

2. Possíveis casos de *update*:

(a) Alteração por *insert* de *hole*:

- i. se B^1 insere *hole* em A1 e B^2 não altera A1, não há conflito e a geometria final de A1 é a estabelecida por B^1 (caso AUIH1). O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;

Figura 4.20: Casos de *delete* de geometrias do tipo área

ii. se B^1 insere *hole* em $A1$ e B^2 também insere *hole* em $A1$ da mesma forma, não há conflito e a geometria final de $A1$ é a nova geometria estabelecida por B^1/B^2 (caso AUIH2);

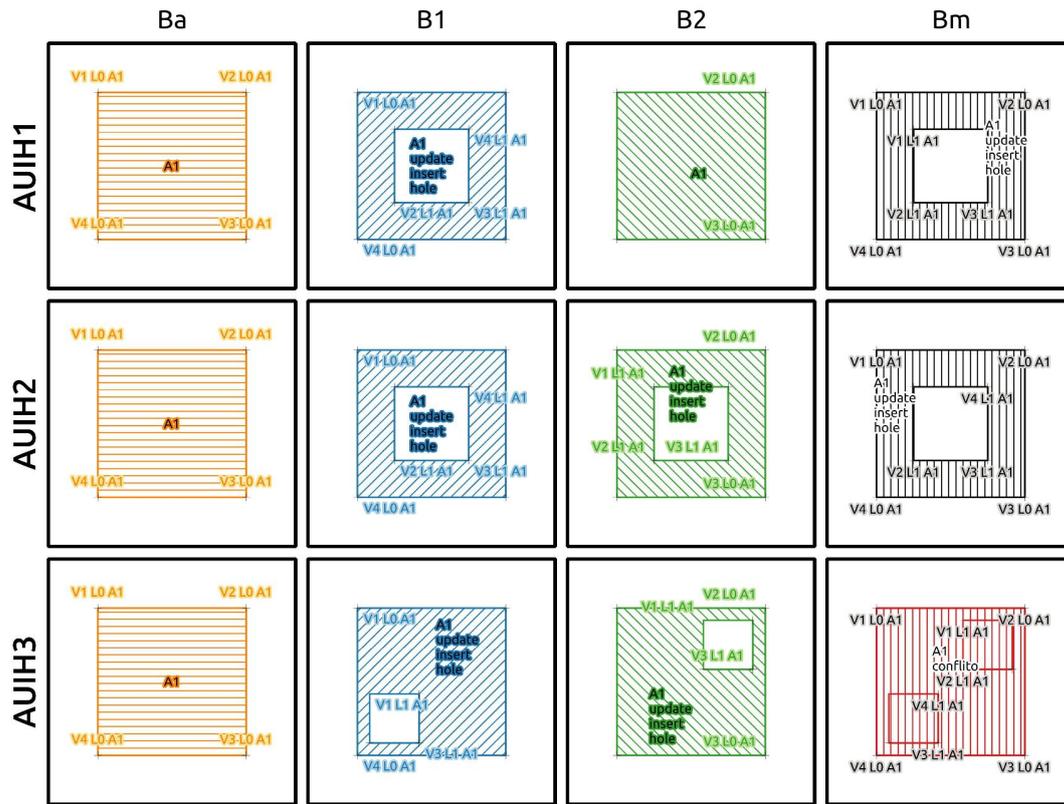
iii. se B^1 insere *hole* em $A1$, e B^2 também insere *hole* em $A1$, porém em posições diferentes, neste caso, tem-se um conflito que precisa ser resolvido no *merge* do versionamento, já que cada *branch* executou uma alteração diferente em $A1$ (caso AUIH3).

(b) Alteração por *delete* de *hole*:

i. se B^1 deleta o *hole* H_k $A1$ e B^2 não altera $A1$, não há conflito e a geometria final de $A1$ é a estabelecida por B^1 (caso AUDH1). O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;

ii. se B^1 deleta o *hole* H_k $A1$ e B^2 também deleta o *hole* H_k $A1$, não há conflito e a geometria final de $A1$ é a nova geometria estabelecida por B^1/B^2 (caso AUDH2);

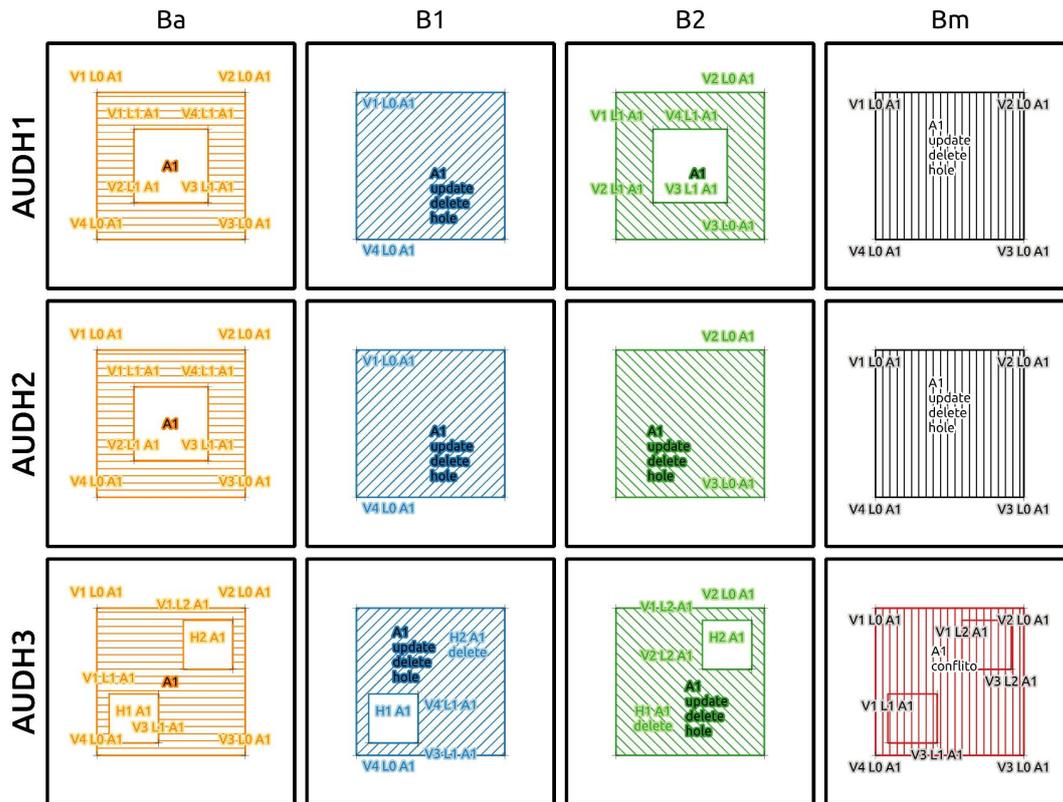
iii. se B^1 deleta o *hole* H_k $A1$ e B^2 também deleta *hole*, porém outro *hole*, por

Figura 4.21: Casos de *insert* de *hole* em geometrias do tipo área

exemplo $H_i A1$, com $k \neq i$, neste caso, tem-se um conflito que precisa ser resolvido no *merge* do versionamento, já que cada *branch* executou uma alteração diferente em $A1$ (caso AUDH3).

(c) Alteração por *insert* de vértices:

- i. se B^1 insere vértices em $L_k A1$ e B^2 não altera $A1$, não há conflito e a geometria final de $A1$ é a estabelecida por B^1 . Esse tipo de caso é possível ocorrer tanto inserção de vértices no *outer ring* (caso AUIV1a) como nos *inner rings* (caso AUIV1b). Ressaltando que o mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 insere vértices em $L_k A1$ e B^2 também insere vértices em $L_k A1$ da mesma forma, não há conflito e a geometria final de $A1$ é a nova geometria estabelecida por B^1/B^2 . Esse tipo de caso é possível ocorrer tanto por inserção de vértices no *outer ring* (caso AUIV2a) como nos *inner rings* (caso AUIV2b);
- iii. se B^1 insere vértices em $L_k A1$ e B^2 também insere vértices em $L_k A1$,

Figura 4.22: Casos de *update* por *delete* de *hole* em geometrias do tipo área

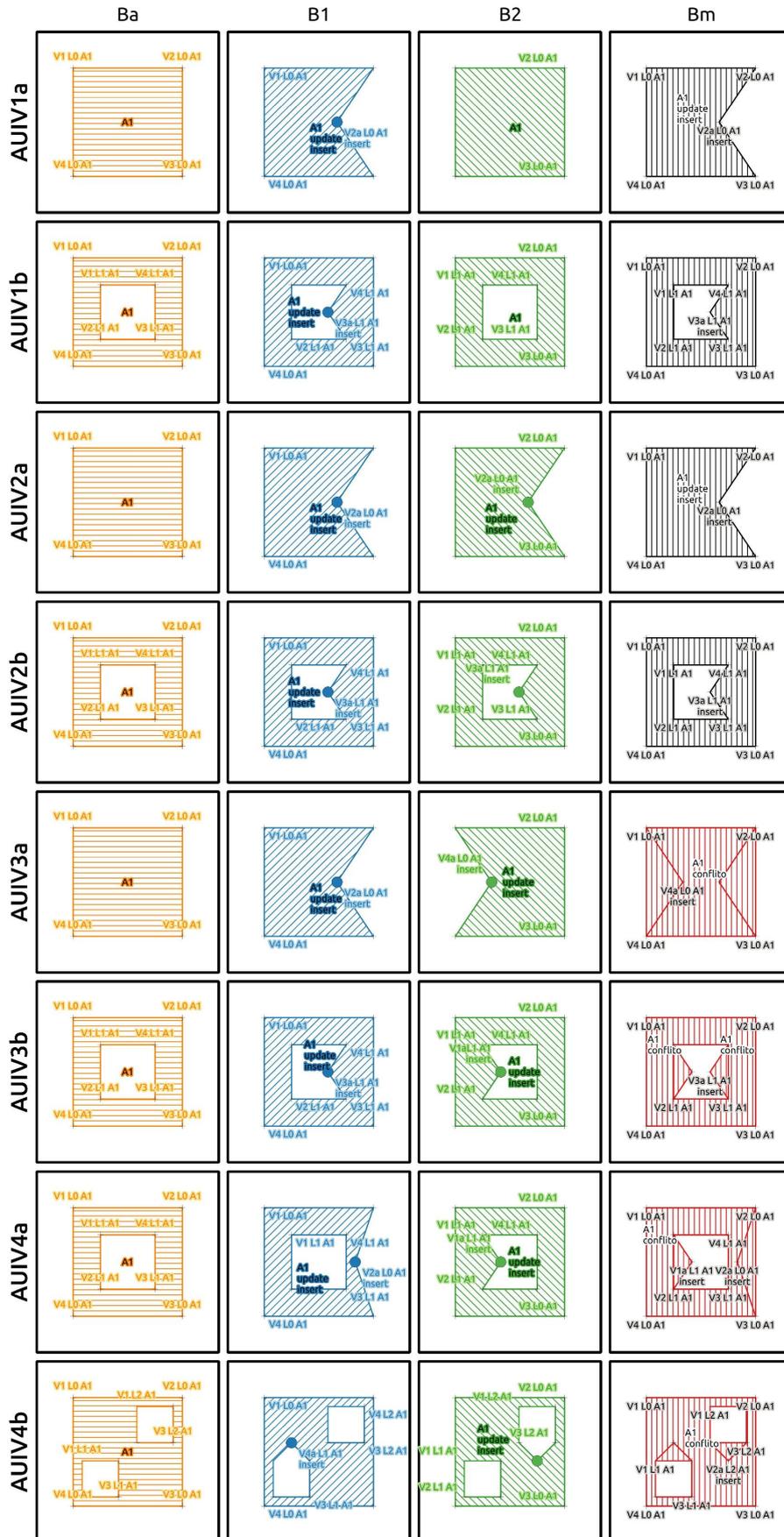
porém de modo diferente, neste caso, tem-se um conflito que precisa ser resolvido no *merge* do versionamento, já que cada *branch* executou uma alteração distinta em A1. Esse tipo de caso é possível ocorrer tanto por inserção de vértices no *outer ring* (caso AUIV3a) como nos *inner rings* (caso AUIV3b);

- iv. se B^1 insere vértices em $L_k A1$ e B^2 insere vértices em $L_i A1$, com $k \neq i$, neste caso, tem-se um conflito que precisa ser resolvido no *merge* do versionamento, já que cada *branch* executou uma alteração distinta em A1. Esse tipo de caso é possível ocorrer tanto por *insert* de vértices entre *outer ring* com *inner ring* (caso AUIV4a) como entre *inner rings* (caso AUIV4b).

(d) Alteração por *delete* de vértices:

- i. se B^1 deleta vértices em $L_k A1$ e B^2 não altera A1, não há conflito e a geometria final de A1 é a estabelecida por B^1 . Esse tipo de caso é possível ocorrer tanto por *delete* de vértices no *outer ring* (caso AUDV1a) como nos *inner rings* (caso AUDV1b);

Figura 4.23: Casos de *update* por *insert* de vértice em geometrias do tipo área

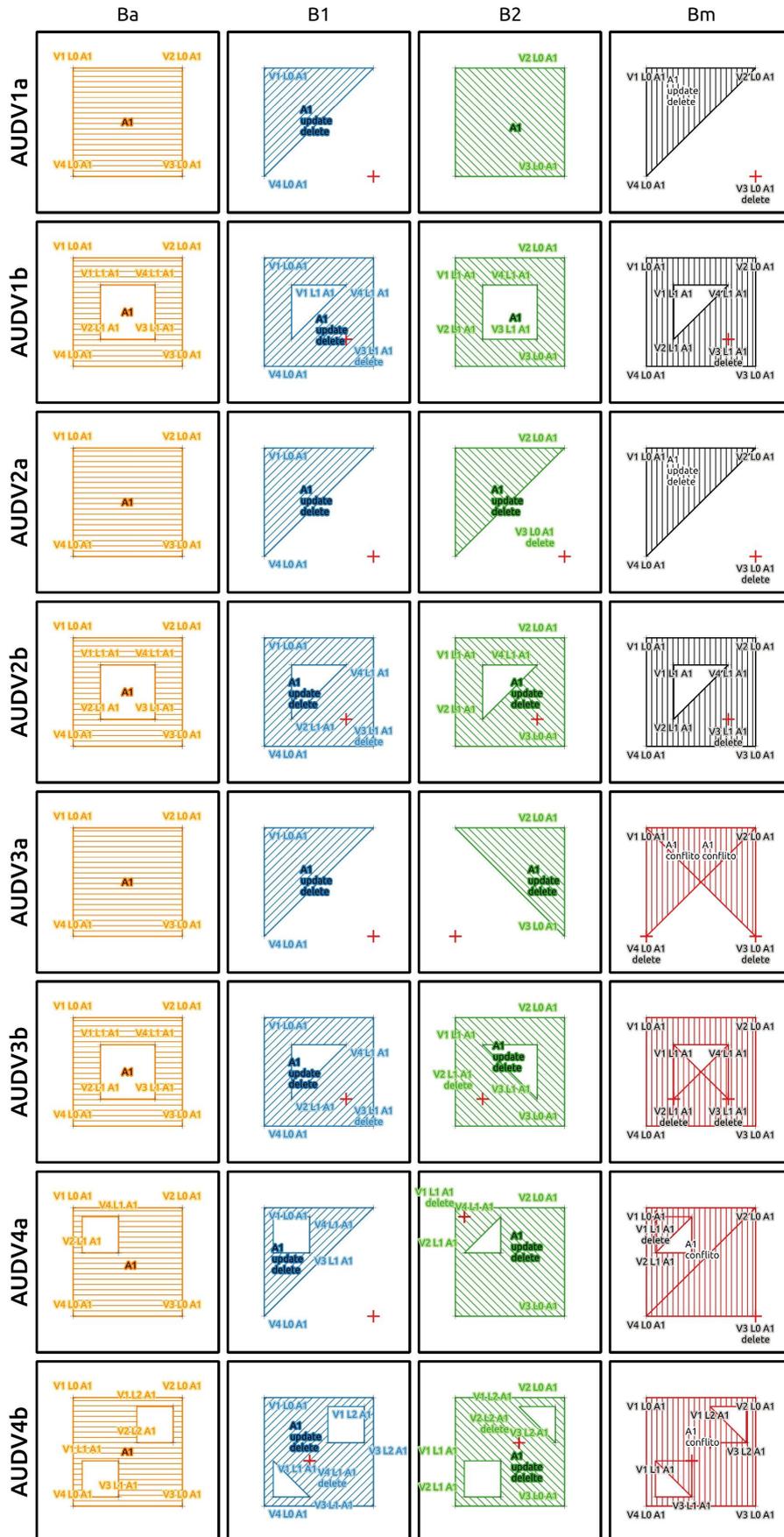


- ii. se B^1 deleta vértices de L_k A1 e B^2 também deleta vértices em L_k A1 da mesma forma, não há conflito e a geometria final de A1 é a nova geometria estabelecida por B^1/B^2 . Esse tipo de caso é possível ocorrer tanto por *delete* de vértices no *outer ring* (caso AUDV2a) como nos *inner rings* (caso AUDV2b);
- iii. se B^1 deleta vértices em L_k A1 e B^2 também deleta vértices em L_k A1, porém de modo diferente, neste caso, tem-se um conflito que precisa ser resolvido no *merge* do versionamento, já que cada *branch* executou uma alteração distinta em A1. Esse tipo de caso é possível ocorrer tanto por *delete* de vértices no *outer ring* (caso AUDV3a) como nos *inner rings* (caso AUDV3b);
- iv. se B^1 deleta vértices em L_k A1 e B^2 também deleta vértices, porém em L_i A1, com $k \neq i$, neste caso, tem-se um conflito que precisa ser resolvido no *merge* do versionamento, já que cada *branch* executou uma alteração distinta em A1. Esse tipo de caso é possível ocorrer tanto por *delete* de vértices entre *outer ring* com *inner ring* (caso AUDV4a) como entre *inner rings* (caso AUDV4b).

(e) Alteração por movimentação de vértices:

- i. se B^1 altera a posição de vértices de L_k A1 e B^2 não altera A1, não há conflito e a geometria final de A1 é a estabelecida por B^1 . Esse tipo de caso é possível ocorrer tanto alteração nos vértices do *outer ring* (caso AUMV1a) como no *inner rings* (caso AUMV1b);
- ii. se B^1 altera a posição de vértices de L_k A1 e B^2 também altera L_k A1 da mesma forma, não há conflito e a geometria final de A1 é a nova geometria estabelecida por B^1/B^2 . Esse tipo de caso é possível ocorrer tanto alteração nos vértices do *outer ring* (caso AUMV2a) como no *inner rings* (caso AUMV2b);
- iii. se B^1 altera a posição de vértices de L_k A1 e B^2 também altera L_k A1, porém de modo diferente, neste caso, tem-se um conflito que precisa ser resolvido no *merge* do versionamento, já que cada *branch* executou uma alteração distinta em A1. Esse tipo de caso é possível ocorrer tanto por alteração de vértices no *outer ring* (caso AUMV3a) como nos *inner rings* (caso AUMV3b);

Figura 4.24: Casos de *update* por *delete* de vértice em geometrias do tipo área

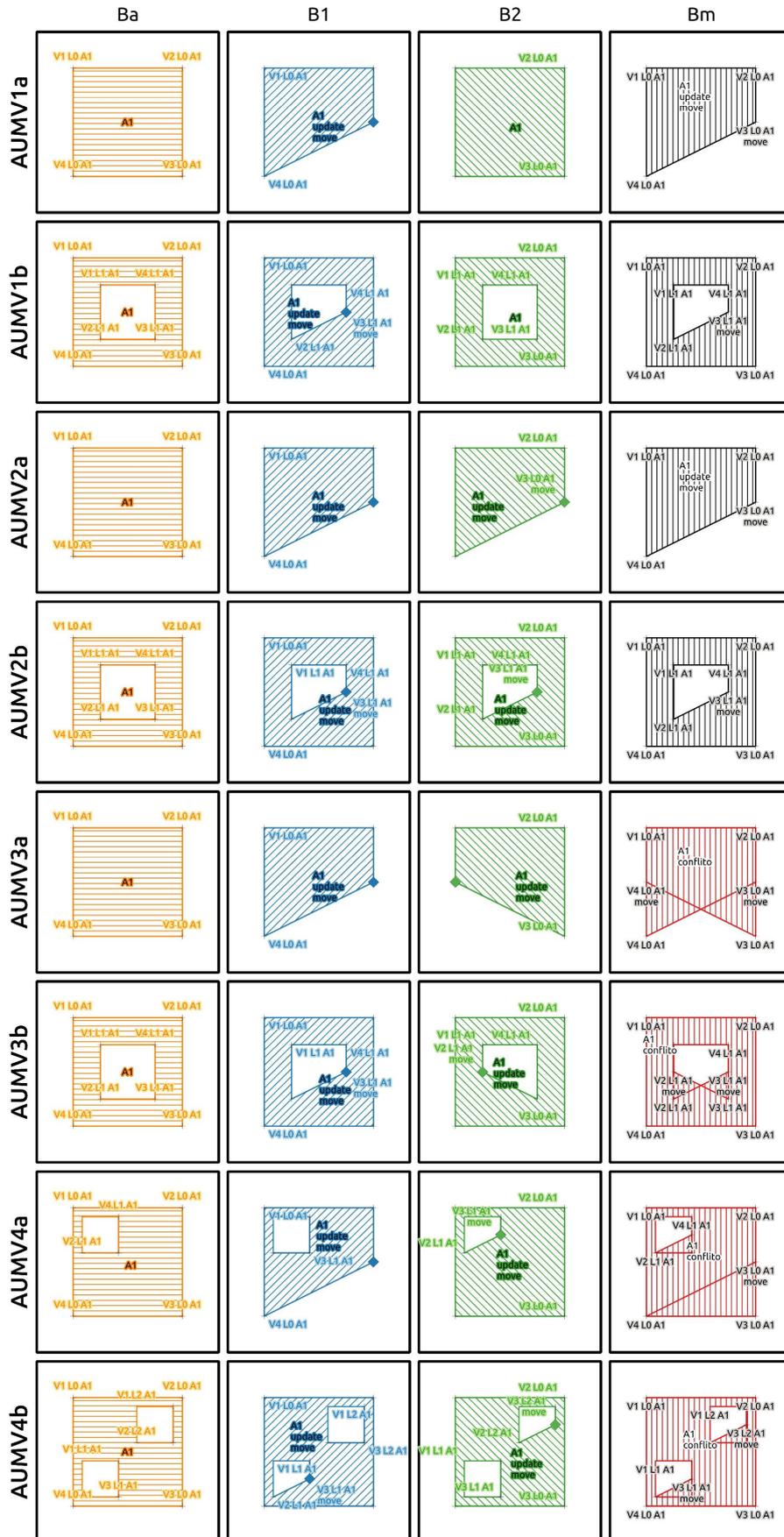


iv. se B^1 altera a posição de vértices de L_k A1 e B^2 também altera A1, porém movendo vértices de L_i A1, com $k \neq i$, neste caso, tem-se um conflito que precisa ser resolvido no *merge* do versionamento, já que cada *branch* executou uma alteração distinta em A1. Esse tipo de caso é possível ocorrer tanto por *insert* de vértices entre *outer ring* com *inner ring* (caso AUMV4a) como entre *inner rings* (caso AUMV4b).

(f) Alteração por *cut*:

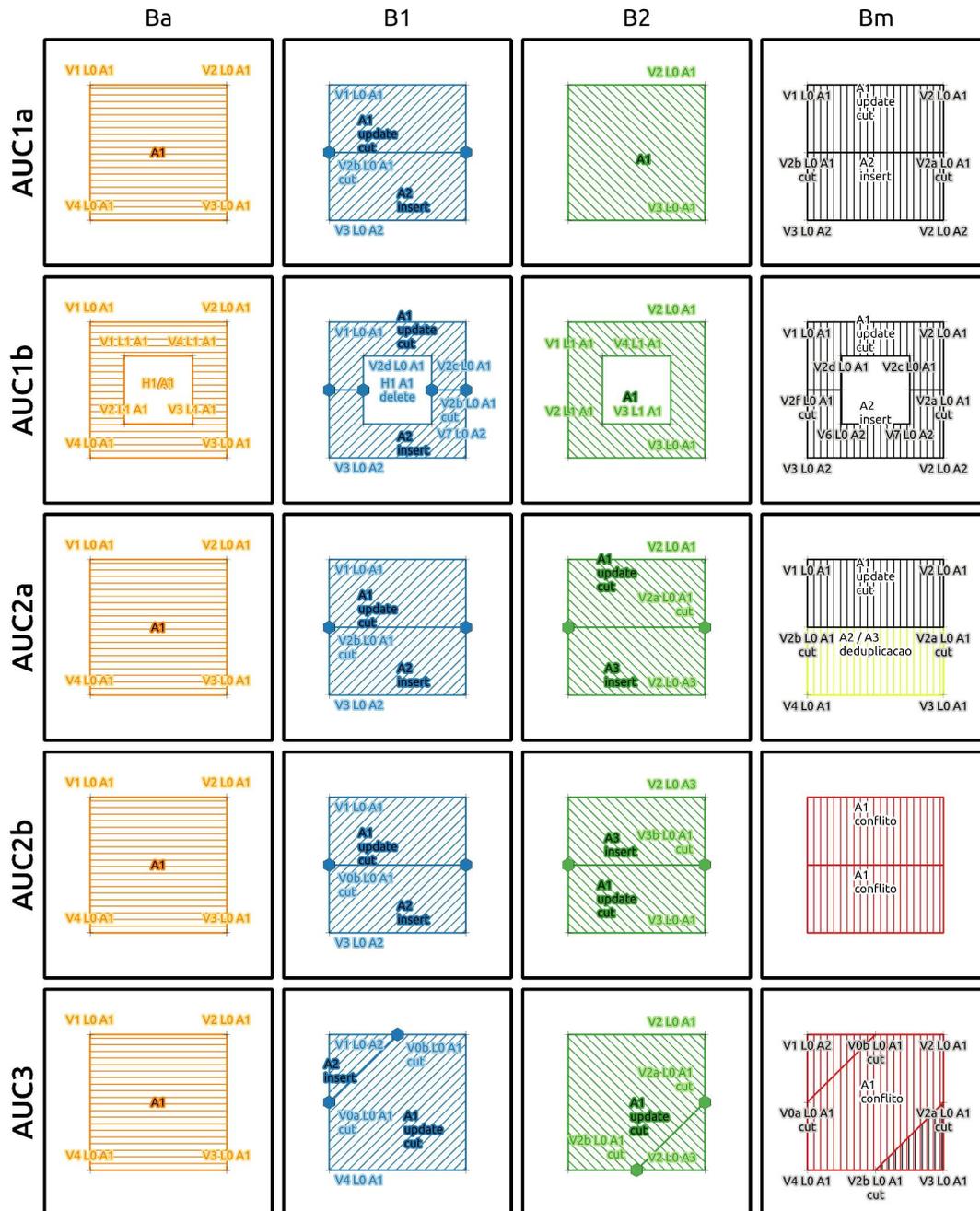
- i. se B^1 corta A1 e B^2 não altera A1, não há conflito e a geometria final de A1 e A2 é a estabelecida por B^1 . Esse tipo de caso é possível ocorrer tanto por corte somente na borda externa da área (caso AUC1a) como por corte envolvendo alguns dos *holes* (caso AUC1b) e ainda com o corte coincidente com alguns dos vértices como quando o corte ocorre entre segmentos. O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 corta A1 e B^2 também corta A1, na mesma posição, pode ocorrer ou não um conflito. Ao cortar uma geometria, uma das partes recebe o id original e a outra recebe um id novo. Assim, pode acontecer do mesmo id ser estabelecido para pedaços diferentes da geometria, o que causa um conflito. A seguir, são apresentados os dois casos: um quando o mesmo pedaço recebe o mesmo id (caso AUC2a), dessa forma não causa conflito, porém gera um caso de inserção de feição duplicada da outra parte a semelhança do caso AI2; e outro quando ocorre “inversão” de id e, por consequência, um conflito (caso AUC2b), e também novos caso AI2, pela inserção de feições duplicadas. Vale ressaltar que esses casos podem ocorrer tanto com o corte somente na borda externa da área como por corte envolvendo alguns dos *holes* e, ainda, com o corte coincidente com algum dos vértices como quando o corte ocorre entre segmentos;
- iii. se B^1 corta A1 e B^2 também corta A1, porém de modo diferente, há conflito, pois há alteração em A1 de modo diferente em cada *branch* (caso AUC3). Esse tipo de caso é possível ocorrer tanto por corte somente na borda externa da área como por corte envolvendo alguns dos *holes* e, ainda, com o corte coincidente com algum dos vértices como quando o

Figura 4.25: Casos de *update* por movimentação de vértices em geometrias do tipo área



corde ocorre entre segmentos.

Figura 4.26: Casos de *update* por *cut* em geometrias do tipo área

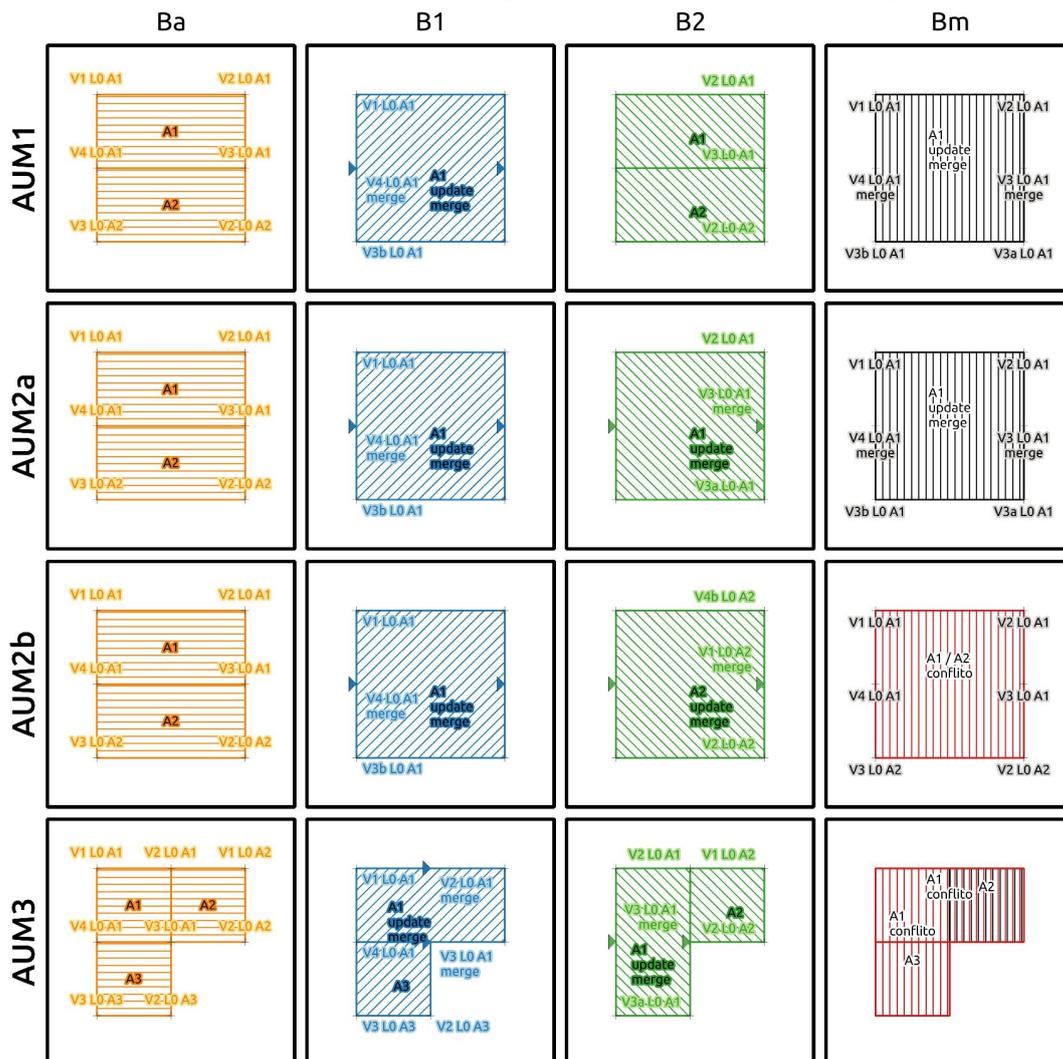


(g) Alteração por operação de GIS do tipo *merge*:

- i. se B^1 faz uma operação de GIS do tipo *merge* de A_1 com A_2 e B^2 não altera A_1 nem A_2 , não há conflito e a geometria final de A_1/A_2 é a estabelecida por B^1 (caso AUM1). O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;

- ii. se B^1 faz uma operação de GIS do tipo *merge* de A1 com A2 e B^2 também faz uma operação de GIS do tipo *merge* de A1 com A2, pode ocorrer ou não um conflito. Ao fazer operação de *merge* de geometrias, a geometria final recebe o id de uma das partes. Assim, pode acontecer o caso da geometria final receber id oriundo de partes diferentes, o que causa um conflito. A seguir, são apresentados os dois casos: um quando ocorre o uso do mesmo de id (caso AUM2a) e outro quando ocorre a “inversão” de id e, por conseguinte, um conflito de versionamento (caso AUM2b);
- iii. se B^1 faz uma operação de GIS do tipo *merge* de A1 com A2, e B^2 também faz uma operação de GIS do tipo *merge*, porém de A1 com A3, há conflito, pois há alteração em A1 de modo diferente em cada *branch* (caso AUM3).

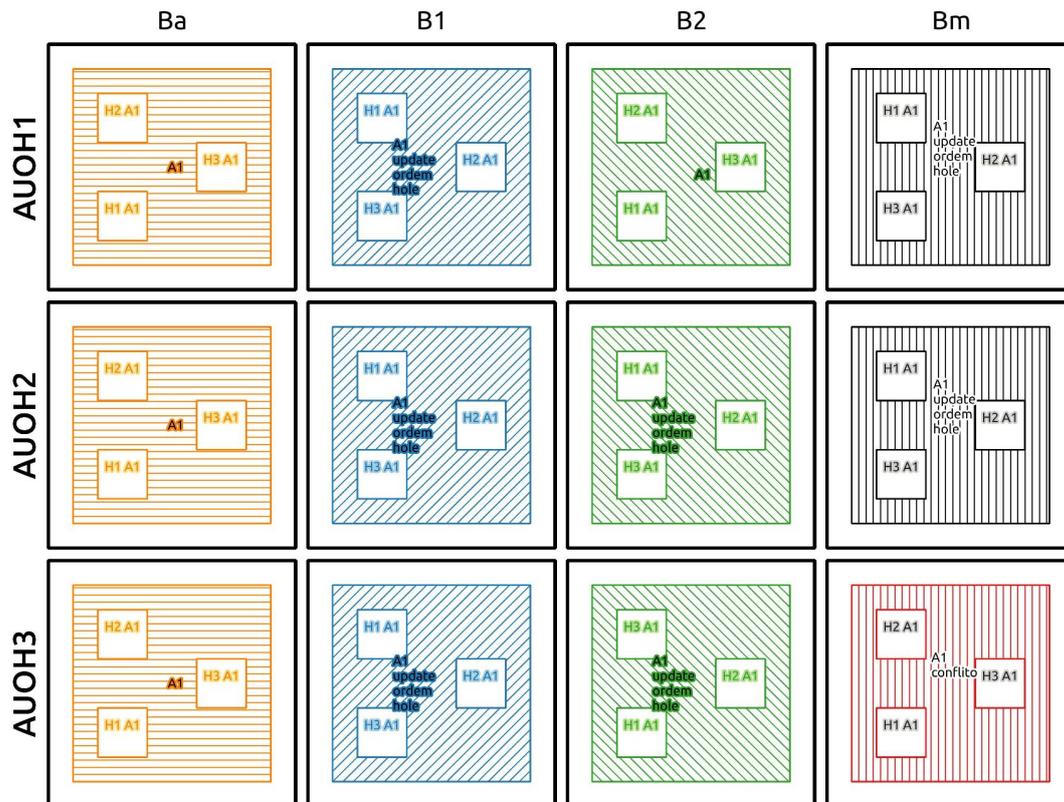
Figura 4.27: Casos de *update* por *merge* de geometrias do tipo área



(h) Alteração de ordenamento dos *holes*:

- i. se B^1 altera a ordem dos *holes* de $A1$ e B^2 não altera $A1$, não há conflito e a geometria final de $A1$ é a estabelecida por B^1 (caso AUOH1). O mesmo raciocínio se aplicaria caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 altera a ordem dos *holes* de $A1$ e B^2 também altera $A1$ da mesma forma, não há conflito e a geometria final de $A1$ é a nova geometria estabelecida por B^1/B^2 (caso AUOH2);
- iii. se B^1 altera a ordem dos *holes* de $A1$ e $A2$ também altera $A1$, porém com outro ordenamento dos *holes*, neste caso, tem-se um conflito que precisa ser resolvido no *merge*, já que cada *branch* executou uma alteração distinta em $A1$ (caso AUOH3).

Figura 4.28: Casos de *update* por iteração no ordenamento de *hole* em geometrias do tipo área



(i) Alteração por rotação de índice de *ring*:

- i. se B^1 promove uma alteração do tipo rotação de índice em L_k $A1$ e B^2 não altera $A1$, não há conflito e a geometria final de $A1$ e $A2$ é a estabe-

lecida por B^1 (caso AUR1). O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;

- ii. se B^1 promove uma alteração do tipo rotação de índice em L_k A1 e B^2 também altera A1 exatamente da mesma forma, não há conflito e a geometria final de A1 é a nova geometria estabelecida por B^1/B^2 (caso AUR2);
- iii. se B^1 promove uma alteração do tipo rotação de índice em L_k A1 e B^2 também altera L_k A1, porém com uma rotação diferente, neste caso, tem-se um conflito que precisa ser resolvido no *merge*, já que cada *branch* executou uma alteração distinta em A1 (caso AUR3).
- iv. se B^1 promove uma alteração do tipo rotação de índice em L_k A1 e B^2 também altera A1, porém em L_i A1, com $k \neq i$, neste caso, tem-se um conflito que precisa ser resolvido no *merge*, já que cada *branch* executou uma alteração diferente em A1 (caso AUR4).

(j) Alteração por *flip* de *ring*:

- i. se B^1 promove uma alteração do tipo *flip* em L_k A1 e B^2 não altera A1, não há conflito e a geometria final de A1 e A2 é a estabelecida por B^1 (caso AUF1). O mesmo raciocínio se aplica caso fossem trocadas as situações apresentadas por B^1 e B^2 , onde a mudança fosse feita somente por B^2 ;
- ii. se B^1 promove uma alteração do tipo *flip* em L_k A1 e B^2 também altera A1 da mesma forma, não há conflito e a geometria final de A1 é a nova geometria estabelecida por B^1/B^2 (caso AUF2);
- iii. se B^1 promove uma alteração do tipo *flip* em L_k A1 e B^2 também altera A1, porém em L_i A1, com $k \neq i$, neste caso, tem-se um conflito que precisa ser resolvido no *merge*, já que cada *branch* executou uma alteração diferente em A1 (caso AUF3).

Figura 4.29: Casos de *update* por rotação de índice em geometrias do tipo área

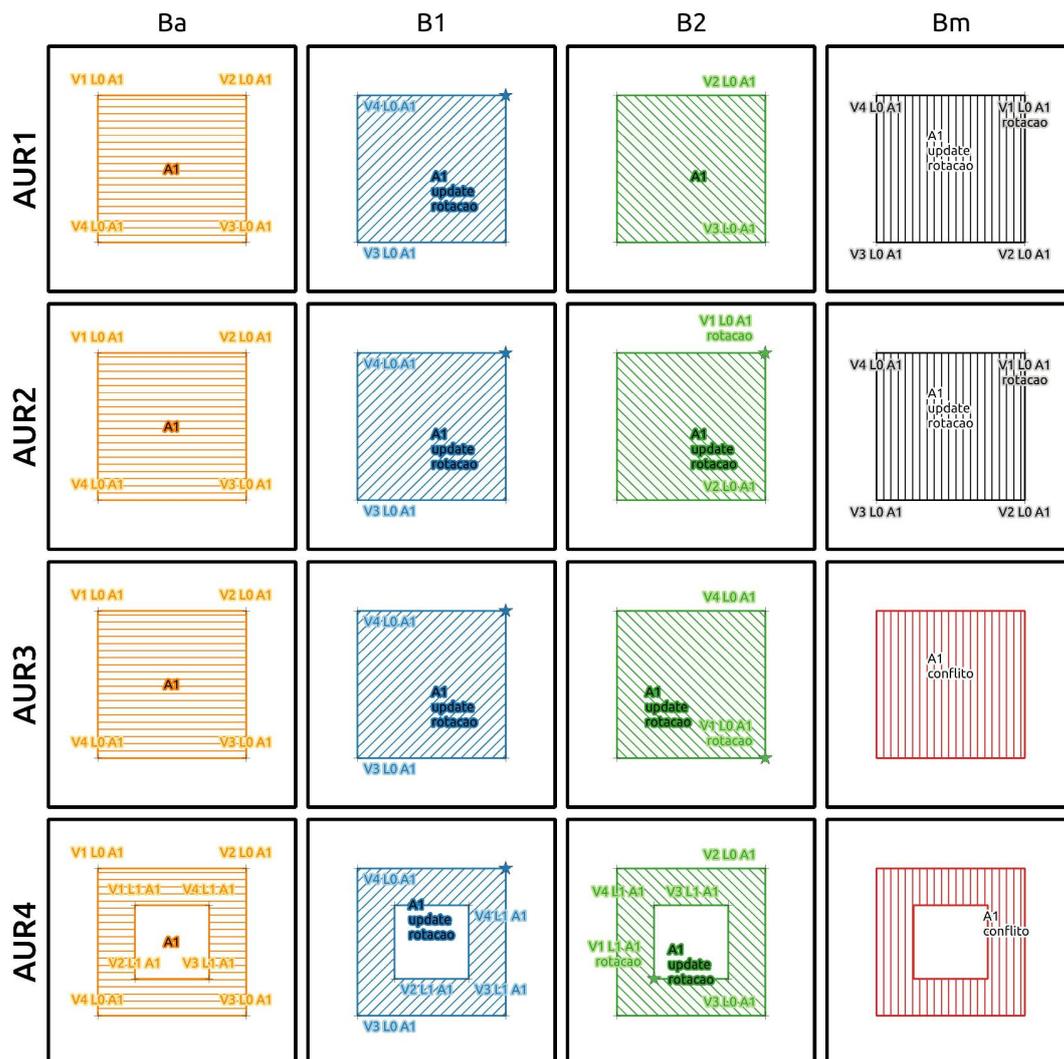
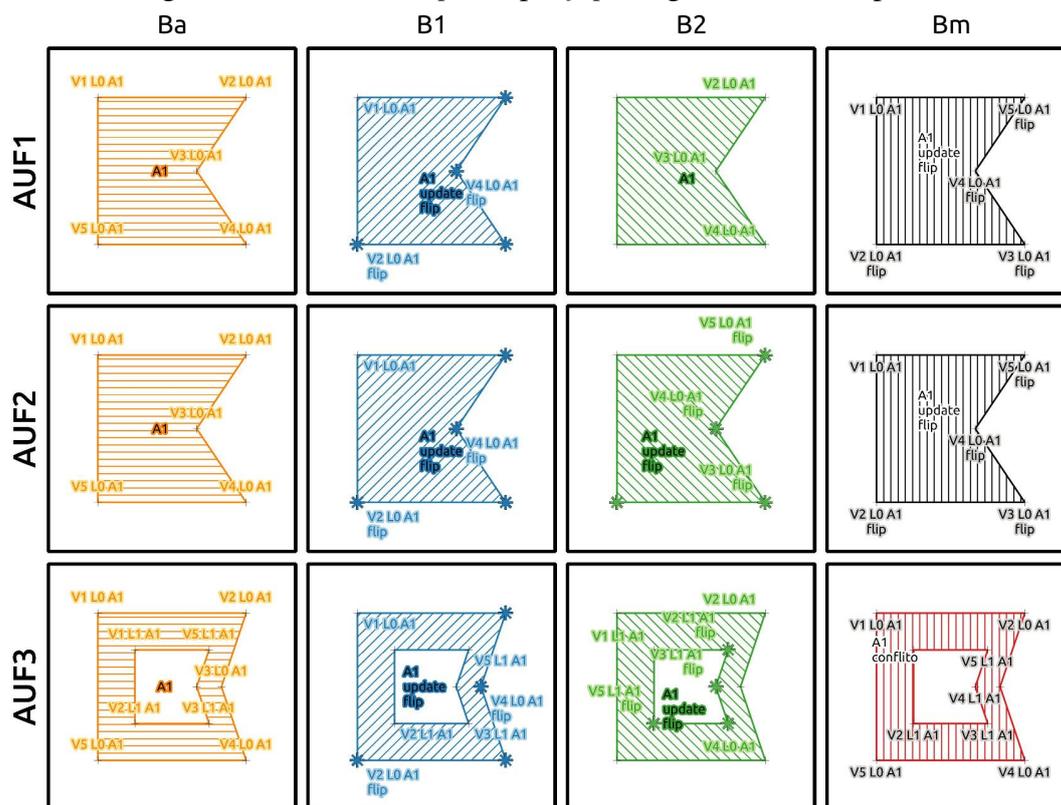


Figura 4.30: Casos de *update* por *flip* em geometrias do tipo área

4.2.4 Conceito geral de versionamento aplicado a dados geográficos

Após a especificação das diversas possibilidades de interação dos dados no momento do versionamento, é possível perceber que alterações unilaterais, que são aquelas feitas somente em um dos *branches*, não resultam em conflito, como é natural em sistemas de versionamento. Alterações feitas nos dois *branches* de modo igual não resultam em conflito. Alterações em que a operação é a mesma, mas que tem diferenças na execução, a exemplo de casos envolvendo *cut* e *merge*, podem gerar conflitos. Após analisar as operações apresentadas é possível sintetizar com a seguinte definição: “*Quaisquer modificações divergentes em vértices são consideradas conflitos*”.

Como demonstrado nos exemplos, todas as situações que ocasionam mudanças nas quais as geometrias não tem igualdade estrita nos vértices, por menor que sejam essas mudanças, ocasionam conflitos. Esse tipo de consideração, denominada igualdade estrita, apesar de ser muito aplicável a diversos tipos de dados, como código de software, *plain text*, entre outros, não permite o uso das potencialidades para dados geográficos. Por esse motivo é apresentado o conceito de *modificação complementar de geometria*, com vistas a atender casos em que as modificações são espacialmente separadas e que nos

sistemas tradicionais de versionamento, ocasionam conflito, de modo desnecessário. A introdução do conceito de modificação complementar de geometria visa atender casos em que os trabalhos dos usuários acabam sendo desconsiderados ou que acabam gerando necessidade de maiores esforços em retrabalhos por falta de tratamento especializado dos dados geográficos. Essa necessidade de retrabalho, focado na resolução de conflitos, na verdade é apenas o não uso do potencial dos dados geográficos.

5 MODIFICAÇÃO COMPLEMENTAR DE GEOMETRIA

O objetivo deste Capítulo é especificar a proposta de modificação complementar de geometria, apresentando um método para sua utilização em dados geográficos. A palavra “complementar” é aqui usada para identificar que as alterações feitas nas geometrias, por parte de usuários, podem ser combinadas, sendo uma alteração complemento da outra, e não conflito, como é tratado nos sistemas atuais. Neste Capítulo é feito o detalhamento dos critérios de modificação complementar, sua aplicação em geometrias dos tipos linha e área, sendo estas as contribuições principais desta dissertação.

5.1 Modificação complementar em geometrias

A presente Seção traz a definição dos critérios que embasaram a modificação complementar de geometrias. Estes critérios são a base da proposta e de como é feita a integração das geometrias em sistemas de versionamento. A partir deles é desenvolvido o método que permite a sua aplicação em sistemas de versionamento de dados.

Os seguintes critérios são adotados para definição de modificação complementar de geometria:

- se um usuário modifica a geometria presente no *branch ancestor*, em um determinado segmento, o outro usuário não pode modificar esse mesmo segmento da geometria;
- se os dois usuários modificam o mesmo segmento, as alterações devem ser iguais.

Para a especificação das proposições, é admitido que segmento da geometria é qualquer subparte da geometria existente entre dois vértices; ou entre o início da geometria e um de seus vértices; ou ainda entre um de seus vértices e o fim da geometria, sendo necessário que os vértices que delimitam o segmento existam nas duas geometrias que serão comparadas. Quaisquer mudanças que se enquadrem nos critérios estabelecidos serão consideradas mudanças complementares.

A definição dos critérios se baseia no entendimento de que as alterações geométricas são intencionais e localizadas, ou seja, quando um usuário altera uma geometria essa alteração é realizada em uma subparte da geometria escolhida pelo usuário. Desta forma, quando dois usuários que alteram uma mesma subparte da geometria de modo distinto, tal alteração foi intencional do ponto de vista de cada usuário, sendo uma diferença de

interpretação entre eles, o que caracteriza um conflito. Caso ambos tenham alterado da mesma forma não há discordância de interpretação.

Após a definição dos critérios que estabelecem as mudanças complementares são definidos os segmentos alterados nas geometrias. Não existe a possibilidade de estabelecer segmentos em geometrias do tipo ponto e por isso não há possibilidade de modificação complementar em pontos. Nas próximas seções, são especificados minuciosamente os segmentos alterados em geometrias dos tipos linhas e áreas e também como deve ser realizada a integração das geometrias que atendam aos critérios de modificação complementar.

5.2 Modificação complementar em geometrias do tipo linha

A presente Seção apresenta a forma de determinar as modificações de segmento em geometria do tipo linha, a aplicação de modificação complementar de geometria em linhas e as considerações a respeito de operações do tipo *flip*. Apresenta ainda os exemplos 1 e 2, onde é demonstrado a aplicação da modificação complementar de geometrias em linha.

5.2.1 Determinando as modificações entre geometrias do tipo linha

O objetivo desta Subseção é apresentar como determinar as modificações feitas na geometria linha, identificando os segmento em que houve mudança. Para isso é feita uma comparação entre os dois *branches*, onde são identificadas as operações feitas na geometria. Esse procedimento é referido como alinhamento de vértices. O Algoritmo 1 especifica o comportamento descrito. Após é apresentada a forma de identificar as operações executadas nas geometrias.

Para estabelecer os segmentos em que houve modificação em uma geometria do tipo linha, é feita a comparação sequencial por índice, vértice a vértice, entre a geometria presente em B^A , e a geometria presente no *branch* posterior, seja ele o *branch* do usuário 1 (B^1) ou o *branch* do usuário 2 (B^2). Percorrendo a geometria presente no *branch ancestor*, vértice a vértice, é feita uma comparação percorrendo os vértices da geometria posterior. Caso o vértice do *ancestor* não seja o mesmo, faz-se uma nova comparação com o vértice subsequente da geometria posterior. Caso o vértice seja o mesmo, passamos para

o próximo vértice do *ancestor* e faz-se a comparação a partir do vértice subsequente da geometria posterior, sem retroceder nos vértices da geometria posterior. Este comportamento é apresentado no Algoritmo 1.

Algorithm 1: Algoritmo para alinhamento de vértices

```

Data:  $[V_0, V_1, \dots, V_k]$  L1  $B^A$  and  $[V_0, V_1, \dots, V_{k'}]$  L1  $B^1$ 
1  $p^1 \leftarrow 0$ ;
2 for  $n^A \leftarrow 0, k$  do
3   for  $n^1 \leftarrow p^1, k'$  do
4     if  $V_{n^A} B^A == V_{n^1} B^1$  then
5        $p^1 \leftarrow p^1 + n^1$ ;    /* vértices correspondentes */
6       Break;

```

Para simplificar a notação, como a comparação é sobre a geometria de mesmo identificador, o identificador é suprimido, sendo referenciado a qual *branch* o vértice pertence. Assim, V_0 L1 do *ancestor branch*, é referido como $V_0 B^A$; e V_0 L1 do *branch* do usuário 1, é referido como $V_0 B^1$. Admite-se ainda que i não é necessariamente igual a i' e que k também não é necessariamente igual a k' , e que $k \geq i \geq j$ e $k' \geq i' \geq j$. As representações esquemáticas das operações de *update* por segmentos são apresentadas nas Figuras 5.1 e 5.2 e os vértices que tem linhas entre B^A e B^1 indicam que são vértices correspondentes, ou sejam, os vértices são iguais.

5.2.1.1 Update por insert de vértice

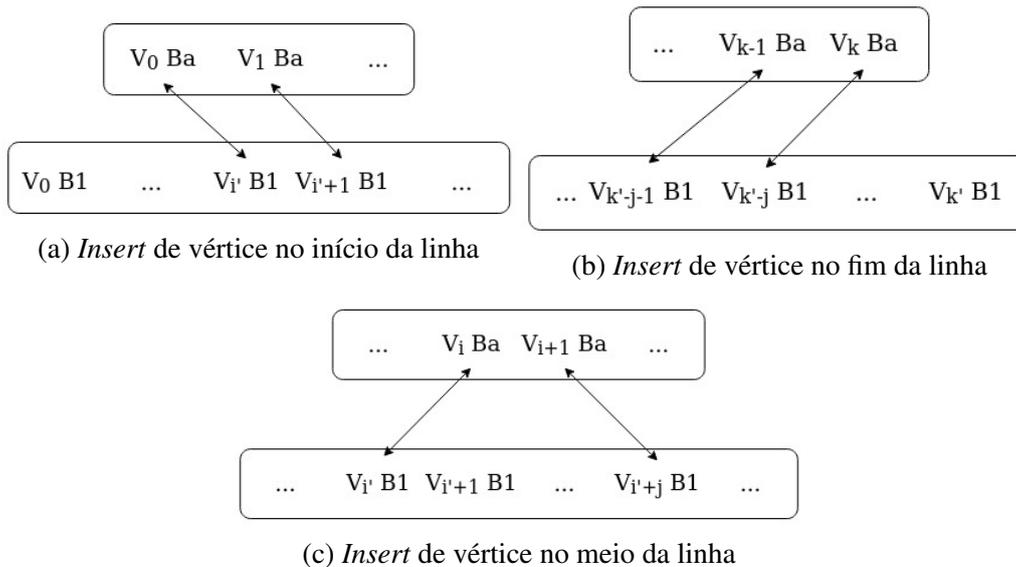
As operações de *update* por *insert* de vértice se caracterizam pela existência de vértices em B^1 sem correspondentes em B^A . Contudo, a posição onde ocorre a inserção de vértices influencia na forma de identificar a operação. Assim, são possíveis três situações de *insert* de vértices:

- *Insert* de vértice(s) no início da linha: a identificação de um ou mais vértices sem correspondentes no início da geometria de B^1 indica que há um *insert* de vértices no início de B^A . Esta configuração demonstra que os vértices de $V_0 B^1$ a $V_{i'-1} B^1$ foram inseridos antes de $V_0 B^A$, promovendo uma alteração no segmento “ $V_0 B^1$ até $V_{i'} B^1$ ”, cujo correspondente em B^A é o segmento “anterior a $V_0 B^A$ ”, conforme ilustrado na Figura 5.1a;
- *Insert* de vértice(s) no fim da linha: a identificação de um ou mais vértices sem correspondentes no final da geometria de B^1 indica que há um *insert* de vértices

no final de B^A . Esta configuração demonstra que os vértices de $V_{k'-j+1} B^1$ a $V_{k'} B^1$ foram inseridos “após $V_k B^A$ ”, promovendo uma alteração no segmento “ $V_{k'-j} B^1$ até $V_{k'} B^1$ ”, cujo correspondente em B^A é o segmento “posterior a $V_k B^A$ ”, conforme ilustrado na Figura 5.1b;

- *Insert* de vértice(s) no meio da linha: a identificação de um ou mais vértices sem correspondentes no meio da geometria B^1 , com B^A tendo vértices sequenciais, indica que há um *insert* de vértices entre dois vértices de B^A . Esta configuração demonstra que os vértices de $V_{i'+1} B^1$ a $V_{i'+j-1} B^1$ foram inseridos entre os vértices $V_i B^A$ e $V_{i+1} B^A$, promovendo uma alteração no segmento “de $V_{i'} B^1$ a $V_{i'+j} B^1$ ”, cujo correspondente em B^A é o segmento “ $V_i B^A - V_{i+1} B^A$ ”, conforme ilustrado na Figura 5.1c;

Figura 5.1: Identificação de *insert* de vértice na comparação de vetores entre *branches*



5.2.1.2 Update por delete de vértices

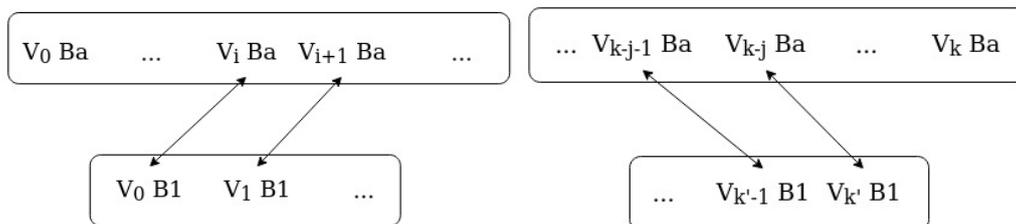
As operações de *update* por *delete* de vértice se caracterizam pela existência de vértices em B^A sem correspondentes em B^1 . Novamente, a posição onde ocorre a alteração dos vértices influencia na identificação da operação. Assim, são possíveis três situações de *delete* de vértices:

- *Delete* de vértice(s) no início da linha: a identificação de um ou mais vértices sem correspondentes no início da geometria de B^A indica que há um *delete* de vértices

no início de B^A . Esta configuração demonstra que os vértices de $V_0 B^A$ a $V_{i-1} B^A$ foram deletados antes do vértice $V_i B^A$, promovendo uma alteração no segmento “ $V_0 B^A$ até $V_i B^A$ ”, ou seja, no segmento “anterior a $V_i B^A$ ”, conforme ilustrado na Figura 5.2a;

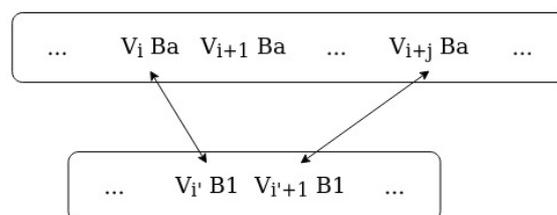
- *Delete* de vértice(s) no fim da linha: a identificação de um ou mais vértices sem correspondentes no final da geometria de B^A indica que há um *delete* de vértices no final de B^A . Esta configuração demonstra que os vértices de $V_{k-j+1} B^A$ a $V_k B^A$ foram deletados depois do vértice $V_{k-j} B^A$, promovendo uma alteração no segmento “ $V_{k-j} B^A$ até $V_k B^A$ ”, ou seja, no segmento “posterior a $V_{k-j} B^A$ ”, conforme ilustrado na Figura 5.2b;
- *Delete* de vértice(s) no meio da linha: a identificação de um ou mais vértices sem correspondentes no meio da geometria B^A , com B^1 tendo vértices sequenciais, indica que há um *delete* de vértices nesse segmento em B^A . Esta configuração demonstra que os vértices “de $V_{i+1} B^A$ a $V_{i+j-1} B^A$ ” foram deletados de B^A , promovendo uma alteração no segmento “ $V_i B^A$ a $V_{i+j} B^A$ ”, conforme ilustrado na Figura 5.2c;

Figura 5.2: Identificação de *delete* de vértice na comparação de vetores entre *branches*



(a) *Delete* de vértice no início da linha

(b) *Delete* de vértice no fim da linha

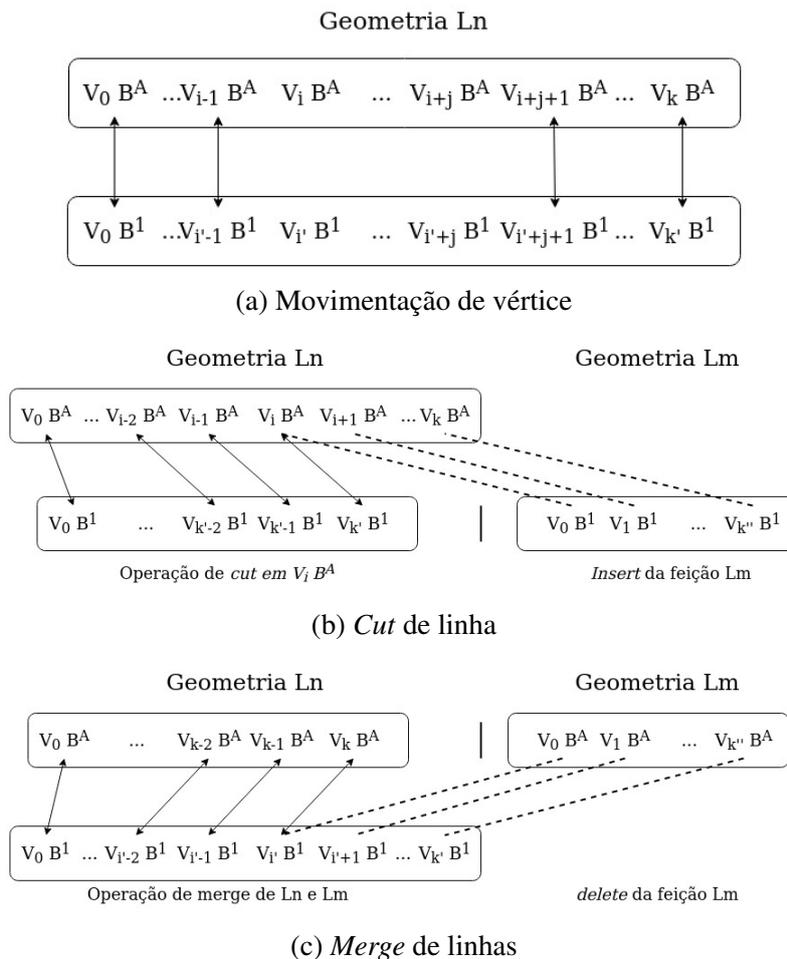


(c) *Delete* de vértice no meio da linha

5.2.2 Update por operações compostas

As operações que podem ser decompostas em operações mais simples não são facilmente identificáveis durante o processo de alinhamento de vértices entre os *branches*. Outro fator que complica é que as operações como *cut* e *merge* envolvem mais de uma feição e, como a determinação das modificações entre *branches* é feita por feição de mesmo identificador, não há como afirmar com exatidão quais operações compostas foram executadas. Ainda que não sejam identificáveis com facilidade, estas operações são especificadas como extensão do modelo de alinhamento de vértices.

Figura 5.3: Identificação de operações compostas em linhas na comparação de vetores entre *branches*



- A Figura 5.3a apresenta uma configuração particular de *delete* de vértices de $V_i B^A$ até $V_{i+j} B^A$, sendo removidos j vértices sequenciais de B^A e com o *insert* de vértices de $V_{i'} B^1$ até $V_{i'+j} B^1$, sendo inseridos j vértices sequenciais em B^1 .

Essa remoção e inserção de mesmo número de vértices, j , entre segmentos correspondentes, sugere que os vértices na verdade foram movidos, causando a perda de correspondência dos vértices entre os *branches*.

- A Figura 5.3b apresenta uma representação expandida, demonstrando uma operação de *cut* no vértice $V_i B^A$. Como é possível observar, a geometria Ln mantém a correspondência entre os vértices até $V_{k'} B^1$. Como a operação de *cut* gera uma nova feição, no caso, geometria Lm , apesar da possível correspondência entre os vértices, como os identificadores são diferentes este fato não acontece. Então, pela perda de correspondência de $V_{i+1} B^A$ ao final de B^A , é entendido que houve uma remoção desses vértices. Ainda, pela não existência da geometria Lm em B^A , é entendido que houve um *insert* dessa geometria em B^1 .
- A Figura 5.3c apresenta uma representação expandida, demonstrando uma operação de *merge* entre as geometrias Ln e Lm . Como a operação de *merge* junta duas feições, apesar da possível correspondência entre os vértices de $Lm B^A$ e $Ln B^1$, a partir de $V_{i'}$ B^1 , pelo fato dos identificadores serem diferente, essa correspondência não acontece. Então, pela perda de correspondência de todos os vértices de $Lm B^A$, é entendido que houve uma remoção dessa geometria, e pela falta de correspondência entre os vértices a partir de $V_{i'+1} B^1$, é entendido que houve o *insert* desses vértices em $Ln B^1$.

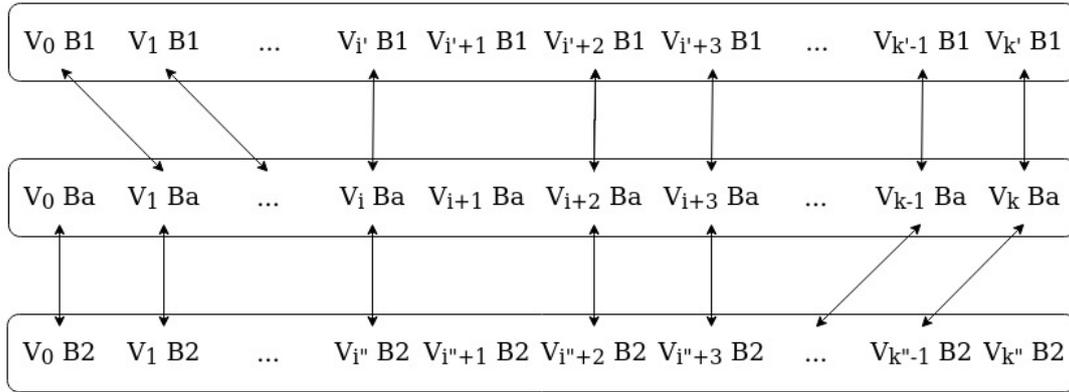
5.2.3 Aplicação de modificação complementar em geometrias do tipo linha

Com base na proposição de modificação complementar da geometrias estabelecida na parte inicial da presente seção, é apresentada sua aplicação para geometrias do tipo linha no contexto de versionamento de dados geográficos. Para tanto, são estabelecidas as mudanças tanto entre B^A e B^1 , quanto entre B^A e B^2 . A Figura 5.4 é usada para as próximas demonstrações.

Fazendo a análise de modificações entre B^A e B^1 tem-se:

- $V_0 B^A$: *delete* do respectivo vértice em B^1 , com alteração do segmento do início até $V_1 B^A$;
- $V_{i+1} B^A$: *delete* do vértice $V_{i+1} B^A$ com o *insert* de $V_{i'+1} B^1$, sendo que a configuração sugere uma movimentação do vértice, com alteração no segmento $V_i B^A$ -

Figura 5.4: Comparação de geometrias entre B^A , B^1 e B^2 para geometrias do tipo linha



$$V_{i+1} B^A - V_{i+2} B^A;$$

Fazendo a análise de modificações entre B^A e B^2 tem-se:

- $V_{i+1} B^A$: *delete* do vértice $V_{i+1} B^A$ com o *insert* de $V_{i''+1} B^2$, sendo que a configuração sugere uma movimentação do vértice, com alteração no segmento $V_i B^A - V_{i+1} B^A - V_{i+2} B^A$;
- $V_{k''} B^2$: *insert* do respectivo vértice em B^2 , após $V_k B^A$, com alteração do segmento de $V_k B^A$ até o fim.

Comparando os segmentos alterados entre B^A-B^1 e B^A-B^2 , tem-se a seguinte situação:

- segmentos de B^A-B^1 :
 1. anterior a $V_0 B^A$: tem-se o *delete* de $V_0 B^A$;
 2. $V_i B^A - V_{i+2} B^A - V_{i+1} B^A$: tem-se a movimentação de $V_{i+1} B^A$ para a posição $V_{i'+1} B^1$.
- segmentos de B^A-B^2 :
 1. $V_i B^A - V_{i+2} B^A - V_{i+1} B^A$: tem-se a movimentação de $V_{i+1} B^A$ para a posição $V_{i''+1} B^2$;
 2. posterior a $V_k B^A$: tem-se o *insert* de $V_{k''} B^2$.

Aplicando os critérios introduzidos como base da modificação complementar de geometrias, tem-se que os segmentos “anterior a $V_0 B^A$ ” oriundo de B^A-B^1 e “posterior a $V_k B^A$ ” oriundos de B^A-B^2 , são modificações que são executadas por somente um dos *branches* e, portanto, atendem ao primeiro critério. Já as modificações em “ $V_i B^A$

- $V_{i+1} B^A - V_{i+2} B^A$ ” são executadas pelos dois *branches* e, de acordo com o segundo critério estabelecido, os dois devem executar a mesma modificação. Por se tratar de uma movimentação em ambos os *branches*, é necessário que $V_{i'+1} B^1$ seja igual a $V_{i''+1} B^2$ para não ser um conflito. Se os vértices forem iguais, tem-se todos os critérios atendidos e as modificações são complementares. Se os vértices não forem iguais, as modificações são um conflito, pois os usuários distintos alteraram mesmo segmento da geometria presente no *branch ancestor* de modo distinto. Caso seja admitido que $V_{i'+1} B^1$ seja igual a $V_{i''+1} B^2$, a geometria final seria $V_1 B^A \dots V_i B^A - V_{i'+1} B^1$ (ou $V_{i''+1} B^2$, pois são equivalentes) - $V_{i+2} B^A - V_{i+3} B^2 \dots V_{k-1} B^A - V_k B^A - V_{k''} B^2$.

5.2.4 Exemplo - Caso 1

Nesta seção, é apresentado o uso de modificação complementar da geometrias para o Caso 1, com base nas seguintes versões da geometria L1:

Ancestor Branch - L1: LINESTRING (252.5 156,253 153.5,253.5
155.5,254.5 154,255.5 156,256 154,257 155,258 153,258
156,256.5 157)

Branch Usuário 1 - L1: LINESTRING (253.5 156.5,252.5 156,253
153.5,254.5 154,255.5 156,256 155,257 155,257 153.5,258
153)

Branch Usuário 2 - L1: LINESTRING (252.5 156,252 154,252.5
155,253 153.5,254.5 154,255.5 154,254.5 156,255.5
156,256 155,257 155,258 153,258 156,256.5 157)

A representação espacial das versões da geometrias são apresentadas na Figura 5.5. A representação do alinhamento dos vértices das versões da geometria L1 é apresentada na Figura 5.6.

Fazendo a análise de modificações entre B^A e B^1 tem-se:

- $V_0 B^1$: *Insert* de vértice, anterior a $V_0 B^A$, com alteração do segmento “anterior a $V_0 B^A$ ”;
- $V_2 B^A$: *Delete* do vértice, com alteração no segmento “ $V_1 B^A - V_2 B^A - V_3 B^A$ ”;

Figura 5.5: Representação espacial do Exemplo 1

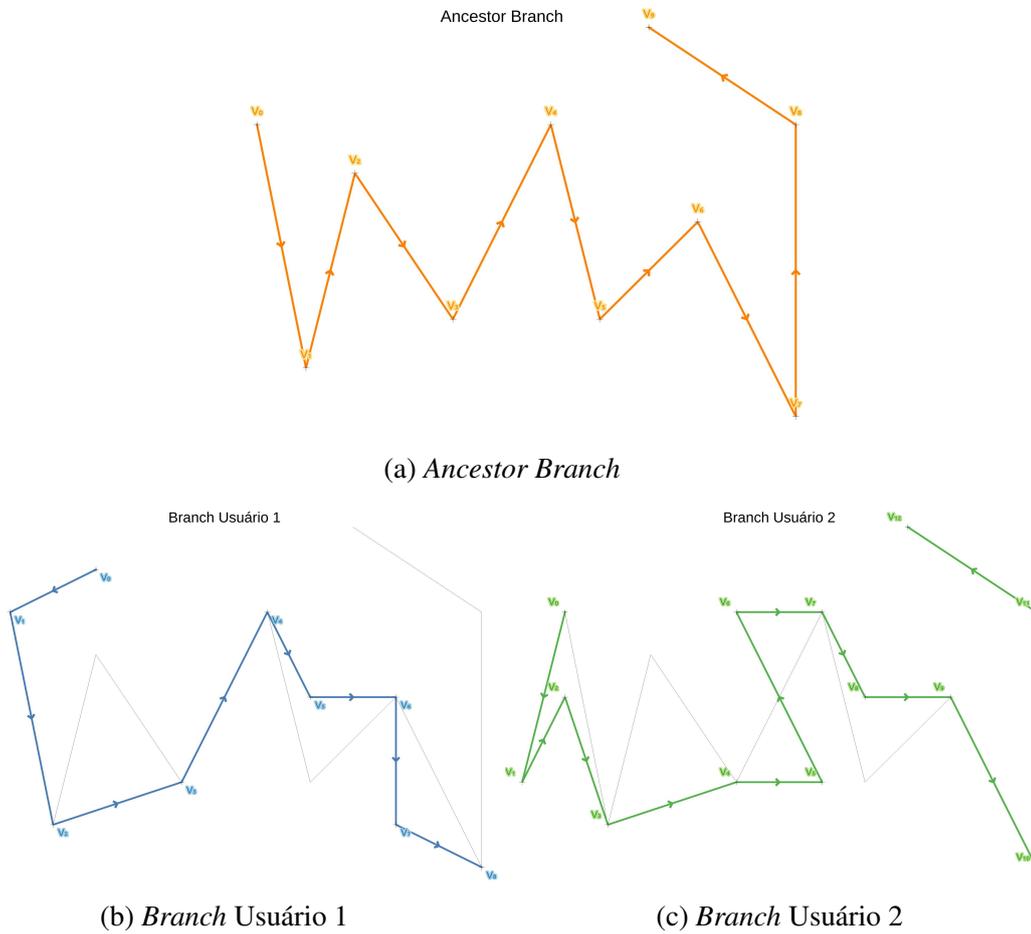
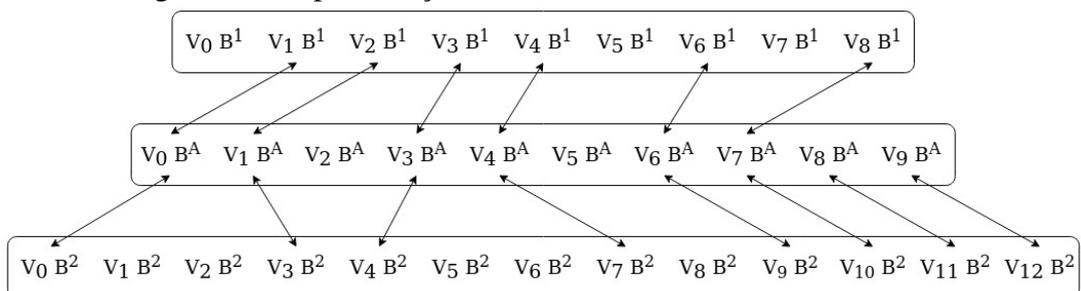


Figura 5.6: Representação do alinhamento dos vértices do Caso 1



- $V_5 B^A$: *Delete* do vértice $V_5 B^A$, combinado com o *insert* de vértice $V_5 B^1$, indicando a movimentação do mesmo. Alteração no segmento “ $V_4 B^A - V_5 B^A - V_6 B^A$ ”;
- $V_7 B^1$: *Insert* de vértice, entre $V_6 B^A$ e $V_7 B^A$, com alteração do segmento “ $V_6 B^A - V_7 B^A$ ”;
- $V_8 B^A$ e $V_9 B^A$: *Delete* dos vértices após $V_7 B^A$, com alteração no segmento “posterior a $V_7 B^A$ ”;

Fazendo a análise de modificações entre B^A e B^2 tem-se:

- $V_1 B^2$ e $V_1 B^2$: *Insert* de vértices, entre $V_0 B^A$ e $V_1 B^A$, com alteração do segmento “ $V_0 B^A - V_1 B^A$ ”;
- $V_2 B^A$: *Delete* do vértice, com alteração no segmento “ $V_1 B^A - V_2 B^A - V_3 B^A$ ”;
- $V_5 B^2$ e $V_6 B^2$: *Insert* de vértices, entre $V_3 B^A$ e $V_4 B^A$, com alteração do segmento “ $V_3 B^A - V_4 B^A$ ”;
- $V_5 B^A$: *Delete* do vértice $V_5 B^A$, combinado com o *insert* de vértice $V_8 B^2$, indicando a movimentação do mesmo. Alteração no segmento “ $V_4 B^A - V_5 B^A - V_6 B^A$ ”;

Comparando os segmentos alterados entre B^A-B^1 e B^A-B^2 , tem-se a seguinte situação:

- segmentos de B^A-B^1 :
 1. anterior a $V_0 B^A$: tem-se o *insert* de $V_0 B^1$;
 2. $V_1 B^A - V_2 B^A - V_3 B^A$: tem-se o *delete* de $V_2 B^A$;
 3. $V_4 B^A - V_5 B^A - V_6 B^A$: tem-se a movimentação de $V_5 B^A$ para a posição do $V_5 B^1$;
 4. $V_6 B^A - V_7 B^A$: tem-se o *insert* de $V_7 B^1$;
 5. $V_7 B^A$ ao final: tem-se o *delete* de $V_8 B^A$ e $V_9 B^A$.
- segmentos de B^A-B^2 :
 1. $V_0 B^A - V_1 B^A$: tem-se o *insert* de $V_1 B^2$ e $V_2 B^2$;
 2. $V_1 B^A - V_2 B^A - V_3 B^A$: tem-se o *delete* de $V_2 B^A$;

3. $V_3 B^A - V_4 B^A$: tem-se o *insert* de $V_5 B^2$ e $V_6 B^2$;
4. $V_4 B^A - V_5 B^A - V_6 B^A$: tem-se a movimentação de $V_5 B^A$ para a posição do $V_8 B^2$;

Aplicando os critérios da modificação complementar de geometrias, tem-se que os segmentos “anterior a $V_0 B^A$ ”, “ $V_6 B^A - V_7 B^A$ ” e “de $V_7 B^A$ ao final da geometria” oriundo de B^A-B^1 ; e os segmentos “ $V_0 B^A - V_1 B^A$ ” e “ $V_3 B^A - V_4 B^A$ ” oriundos de B^A-B^2 , são modificações que são executadas por somente um dos *branches* e, portanto, atendem ao primeiro critério. Já as modificações “ $V_1 B^A - V_2 B^A - V_3 B^A$ ” e “ $V_7 B^A$ ao final” são executadas pelos dois *branches*. Analisando o primeiro segmento, é possível perceber que se trata da remoção do vértices $V_2 B^A$ em ambos e, sendo igual em ambos, atende ao segundo critério. Analisando o segundo segmento, é possível perceber que se trata da movimentação em ambos, e observando o vértice $V_5 B^1$ (256 155) e o vértice $V_8 B^2$ (256 155), é possível perceber que se trata do mesmo vértice, portanto, também atendendo ao segundo critério. Assim, a respeito da geometria apresentada, suas versões são um caso de modificação complementar de geometrias, sendo possível sua integração. A geometria final seria L1: $V_0 B^1 - V_0 B^A - V_1 B^2$ e $V_1 B^2 - V_3 B^A - V_5 B^2 - V_6 B^2 - V_4 B^A - V_5 B^1$ (ou $V_8 B^2$, pois são equivalentes) - $V_6 B^A - V_7 B^1 - V_7 B^A$. A definição correspondente em WKT é `LINSTRING (253.5 156.5,252.5 156,252 154,252.5 155,253 153.5,254.5 154,255.5 154,254.5 156,255.5 156,256 155,257 155,257 153.5,258 153)` e a Figura 5.7 traz a representação espacial da geometria final.

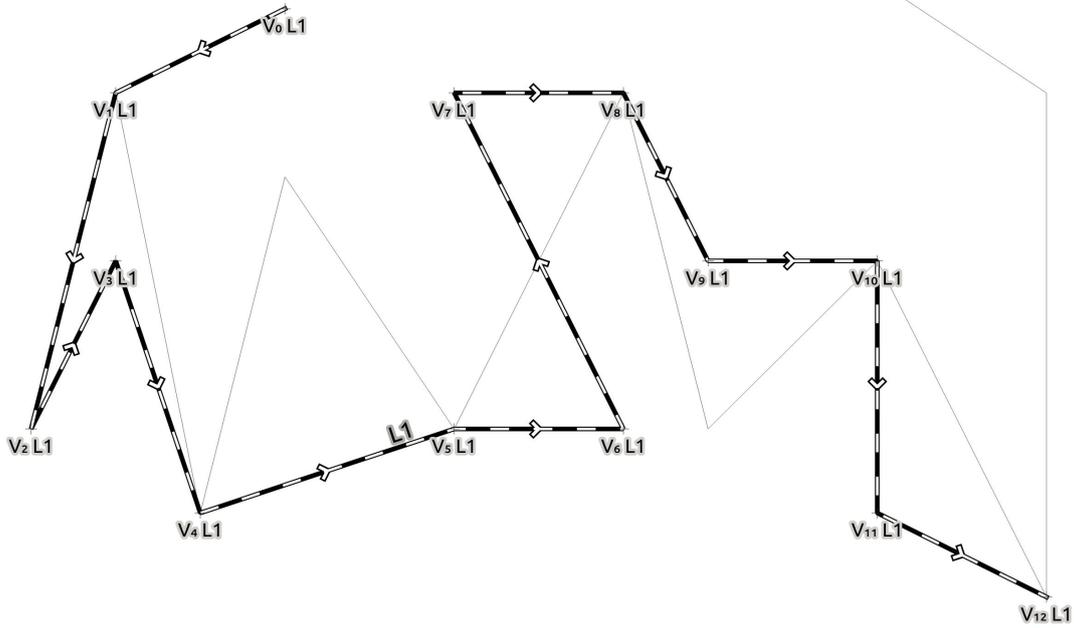
5.2.5 Operações do tipo *flip* em geometrias do tipo linha

A presente Subseção apresenta a forma de determinar as operações do tipo *flip* e seu tratamento em relação modificação complementar de geometrias. Apresenta também o exemplo 2, no qual é feito a aplicação da modificação complementar de geometrias quando é identificada uma operação to tipo *flip*.

5.2.5.1 Determinando operação de *flip*

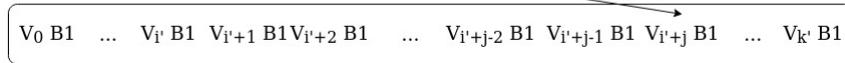
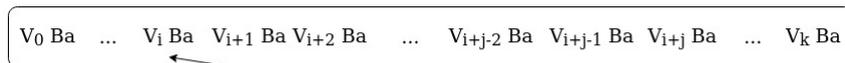
Além das operações já apresentadas para as geometria do tipo linha, existe ainda a operação de *flip*, que é uma operação que atua em toda a geometria, de modo global, e que promove uma inversão completa do ordenamento dos vértices da linha. Alterações

Figura 5.7: Geometria integrada - Caso 1
Merged Branch - Geometria integrada

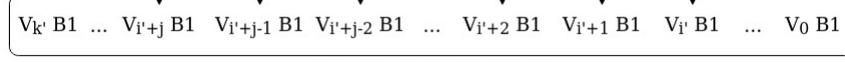
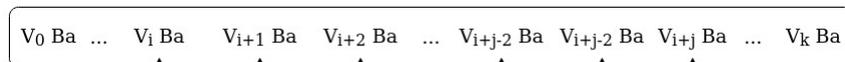


do tipo *flip* são caracterizadas pela existência de correlação de um único vértice entre as geometrias de B^A e B^1 e pela existência de mais de uma correlação de vértices entre a geometria B^A e o ordenamento inverso da geometria de B^1 , conforme apresentado na Figura 5.8.

Figura 5.8: Operação de *flip*



(a) Característica de possível operação *flip*



Inversão do ordenamento de B^1

(b) Confirmação de operação *flip*

Para que o método modificação complementar de geometria funcione corretamente, deve-se identificar operações do tipo *flip* primeiro, para só depois prosseguir na identificação de segmentos alterados. Dessa forma, uma vez identificada a operação de

flip na geometria, a mesma é marcada como uma geometria inversa e a geometria B^1 segue invertida para os passos seguinte, na identificação de segmentos alterados.

5.2.5.2 Exemplo - Caso 2

Para exemplificar a operação *flip*, é apresentado o Caso 2, cujas versões da geometria L1 são apresentadas a seguir:

Ancestor Branch - L1: LINESTRING (252.5 156,253 153.5,253.5
155.5,254.5 154,255.5 156,256 154,257 155,258 153,258
156,256.5 157)

Branch Usuário 1 - L1: LINESTRING (253.5 156.5,252.5 156,253
153.5,254.5 154,255.5 156,256 155,257 155,257 153.5,258
153)

Branch Usuário 2 - L1: LINESTRING (256.5 157,258 156,258 153,257
155,256 155,255.5 156,254.5 156,255.5 154,254.5 154,253
153.5,252.5 155,252 154,252.5 156)

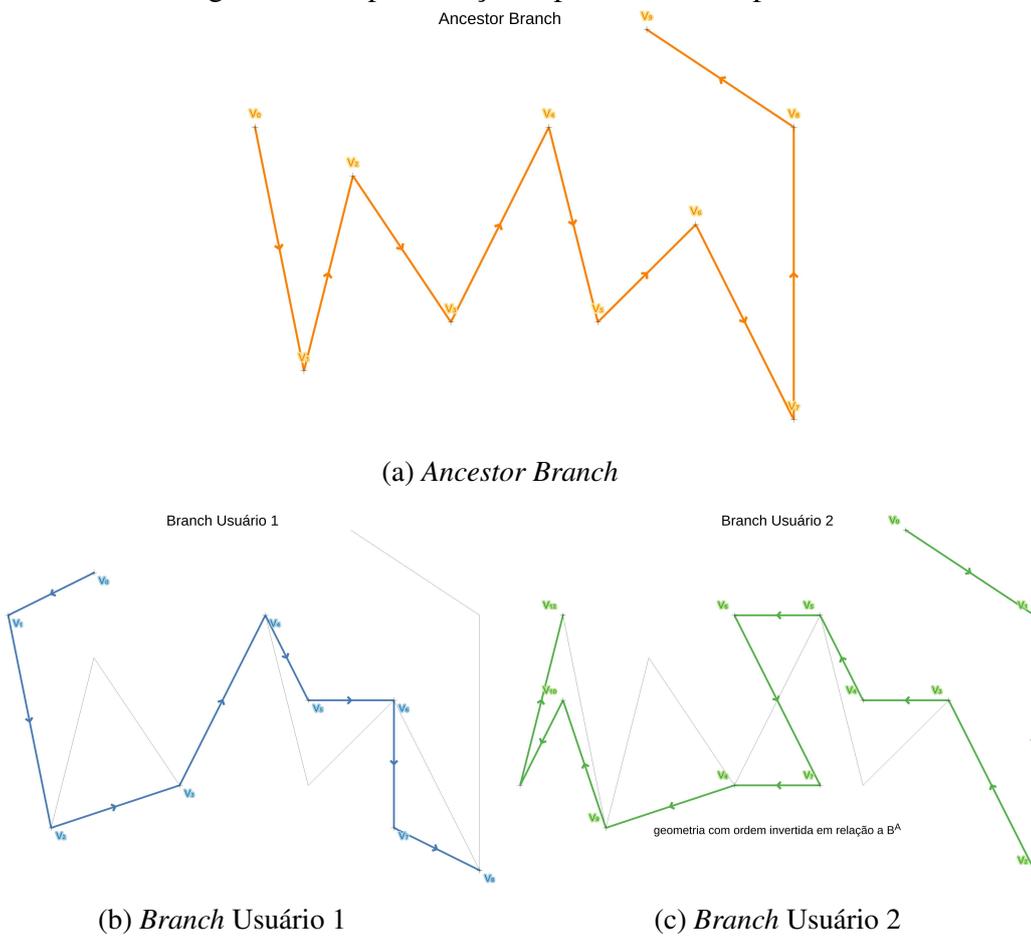
Observando as geometrias do caso 2, as versões de B^A e B^1 são exatamente as mesmas versões do exemplo do caso 1. Já a nova versão de B^2 é a geometria B^2 da caso 1 em ordem inversa. A representação espacial das versões da geometrias são apresentadas na Figura 5.9.

Realizando o método de alinhamento de vértices entre B^A e B^1 tem-se a identificação das mesmas alterações do Caso 1. Realizando alinhamento de vértices entre B^A e B^2 , nota-se um importante acréscimo nas alterações, *com a identificação de operação de flip em B^2* e, continuando o método de identificação de alterações de segmento, identifica-se que o restante das alterações é exatamente igual ao Caso 1 também.

5.2.5.3 Operação de flip em modificação complementar de geometrias

Na Subseção 4.1.2 foi apresentado que operações de *flip* podem ter ligação com característica física do mundo real relacionadas ao direcionamento. Porém, nem todas as camadas tem intrínseca essa necessidade de direcionamento. Tomando como exemplo a classe “Delimitacao_Fisica”, presente na ET-ETGV (DIRETORIA DE SERVIÇO GEOGRÁFICO, 2017), que engloba objetos como cercas, muros, gradis, muretas. Para esse tipo de objeto, não há necessidade de rigor no direcionamento durante a construção

Figura 5.9: Representação espacial do Exemplo 2



da geometria, pois a direção não se traduz em nenhum atributo real do objeto. Contudo, mesmo não sendo necessária, a combinação de operações, pode ocasionar essa alteração de maneira indireta na feição, o que é tratado como uma mudança na feição pelo método proposto. Assim, durante o versionamento, é necessário um tratamento diferenciado para cada caso.

Com relação a modificação complementar de geometria, as operações de *flip* necessitam de tratamento específico. Como visto, há casos que é necessário o rigor no tratamento do direcionamento (comportamento padrão) e há casos onde o rigor não é necessário (comportamento especial). Dessa forma, é necessário que o usuário informe sobre o tipo de tratamento específico que deve ser adotado para cada classe por parte do método. O tratamento de operações de *flip* em modificação complementar de geometrias seguem o seguinte critério:

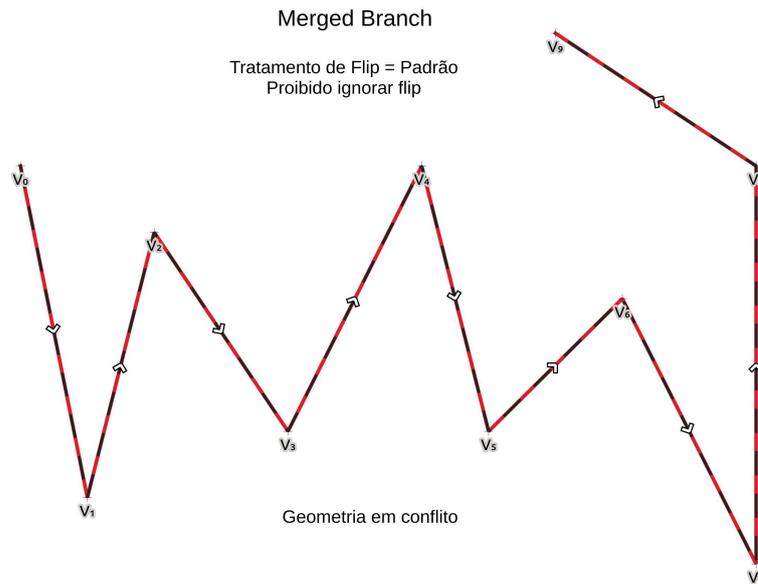
- comportamento padrão: nesse caso, a direção da linha tem importância e, em situação de modificação complementar de geometrias, os dois *branches* têm que atuar da mesma forma com relação a operação de *flip* para que não haja conflito de versionamento. Caso um *branch* apresente um operação de *flip* e o outro não, tem-se um conflito de versionamento.
- comportamento especial: nesse caso, a direção da linha não tem importância e, em situação de modificação complementar de geometrias, deve-se descartar a indicação de operação de *flip*, caso um dos *branches* tenha executado esta operação. No comportamento especial não há situação de conflito por operação de *flip*.

Retomando o exemplo do caso 2, pode-se agora especificar duas possibilidades de integração da geometria, a depender de como deve ser o comportamento previsto para a classe. A primeira possibilidade é o comportamento padrão para modificação complementar, sub-caso 2a, sem possibilidade de ignorar a operação de *flip*. Analisando as operações identificadas, tem-se que entre B^A e B^2 foi identificada uma operação de *flip*, sem operação correspondente em B^1 . Esse fato é suficiente para impedir a integração da geometria e apontá-la como um conflito para resolução manual por parte do operador. A representação espacial do sub-caso 2a é apresentada na Figura 5.10a.

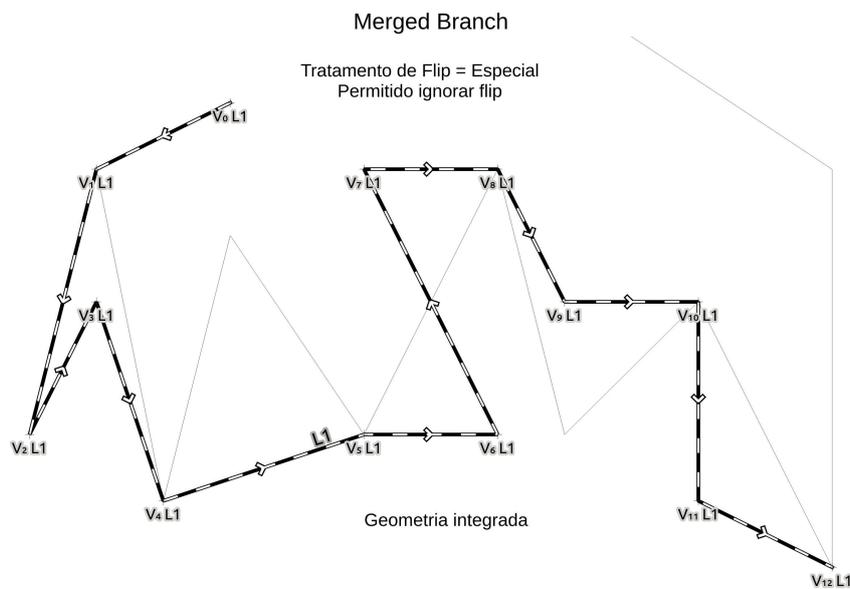
A segunda possibilidade é o comportamento especial para modificação complementar, sub-caso 2b, no qual é possível ignorar a operação de *flip*. Neste comportamento, ao ignorar a operação de *flip* que ocorre em B^2 , todas as alterações remanescentes são as mesmas apresentadas no caso 1. Com isso, é possível integrar a geometria L1,

cuja definição final em WKT é LINESTRING (253.5 156.5,252.5 156,252 154,252.5 155,253 153.5,254.5 154,255.5 154,254.5 156,255.5 156,256 155,257 155,257 153.5,258 153), sendo sua representação espacial apresentada na Figura 5.10b.

Figura 5.10: Representação espacial dos Sub-casos do Exemplo 2



(a) Comportamento padrão para integração de operação *flip*



(b) Comportamento especial para integração de operação *flip*

5.3 Modificação complementar em geometrias do tipo área

A presente Seção apresenta a forma de determinar as modificações de segmento em geometria do tipo área, demonstrando como determinar o pareamento de *inner rings* das áreas, além de apresentar a aplicação de modificação complementar de geometrias para as áreas. Apresenta também o exemplo 3, no qual é feito a aplicação da modificação complementar de geometrias para área. Ainda são feitas as considerações a respeito de alterações do tipo *flip* e rotação de índice de *ring*.

5.3.1 Determinando as modificações entre geometrias área

Como visto na Seção 4.1.3, as áreas são composta por uma linha externa e zero, uma ou muitas linhas internas, que tem a propriedade de serem linhas fechadas por definição, sendo então chamadas de *rings*. Então, por serem linhas, todas as operações definidas para as linhas também são válidas para os *rings*, que delimitam as áreas na hora de determinar as modificações.

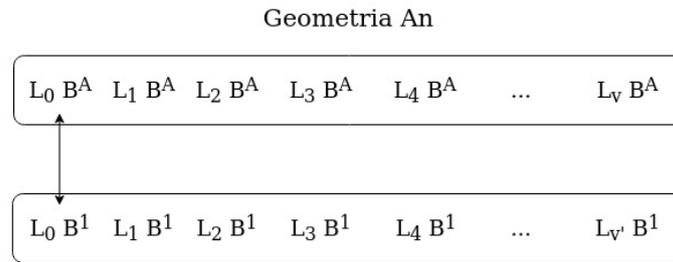
Entretanto, existem propriedades particulares para áreas e para os *rings* que a compõem. Uma delas é a existência de *inner rings*, que formam os *holes*. Os *holes*, em sentido físico, são subregiões em que a cobertura expressa pela delimitação externa da área não se verifica e, por isso, são “furos” na área em posições específicas. Na implementação do dado geográfico, por exemplo o WKT, existe o ordenamento dos *holes* e esse ordenamento não tem relacionamento com as propriedades das áreas no mundo físico. Esse tipo de ordenamento pode ocasionar conflitos. Assim, de forma a garantir a relação entre os *holes* das diferentes versões é necessário buscar correspondências que envolvam suas posições espaciais. Para determinar o par de *holes* que será comparado é necessário utilizar uma medida de similaridade. Existem diversas medidas de similaridades que podem ser aplicadas em polígonos, como a distância de *hausdorff* e interseção sobre união. O método interseção sobre união foi escolhido por não somente considerar o anel exterior, como também a área interior do polígono. Dessa forma, é usado como critério de correspondência o maior valor da interseção dividido pela união entre *holes* de diferentes *branches*, fazendo o cálculo de todos *holes* de B^A com todos os *holes* de B^1 .

Novamente, para simplificar a notação, como a comparação é sobre a geometria de mesmo identificador, o identificador é suprimido, sendo referenciado a qual *branch* a geometria pertence. Assim, $L_i A1$ do *branch ancestor*, é referido como $L_t B^A$; e $L_{t'} A1$

do *branch* do usuário 1, é referido como $L_{t'} B^1$. Com t e t' variando de 0 ao número máximo de *inner rings* ($= v$) das respectivas geometrias. Quando t (ou t') > 0 , $H_t B$ é o hole definido por $L_t B$. Cabe lembrar que cada $L_t B$ é composto por vértices, que são referidos como $V_i L_t B$, com i podendo variar de 0 ao “(número de máximo de vértices de L_i) – 1 = k ”.

$$f_{tt'} = \frac{H_t B^A \cap H_{t'} B^1}{H_t B^A \cup H_{t'} B^1} \quad \forall \quad 1 \leq t \leq v, \quad 1 \leq t' \leq v' \quad \Leftrightarrow \quad H_t B^A \cap H_{t'} B^1 \geq 0 \quad (5.1)$$

Figura 5.11: Exemplo de pareamento inicial de *hole* em geometrias áreas



Para exemplificar o pareamento dos *holes*, é usada a Figura 5.11. A correspondência entre $L_0 B^A$ e $L_0 B^1$ existe pela definição do dado geográfico, não havendo outro tipo de pareamento possível para $L_0 B$. Já os *inner rings* devem ser pareados de acordo com a equação 5.1, que define $f_{tt'}$. A Tabela 5.1 apresenta valores de $f_{tt'}$. Como é possível verificar, $H_1 B^A$ possui $f > 0$ com $H_1 B^1$ e $H_2 B^1$, e $f = 0.0$ para os demais *holes* de B^1 . Já $H_2 B^A$ possui $f > 0$ com $H_2 B^1$ e $H_3 B^1$, e, também, $f = 0.0$ para os demais *holes* de B^1 . Já $H_3 B^A$ possui $f = 0$ para todos os *holes* de B^1 . Quanto a $H_4 B^A$, este possui $f = 1$ com $H_4 B^1$ e $f = 0$ para os demais *holes* de B^1 , situação que se repete de $H_5 B^A$ até $H_v B^A$, possuindo correspondência de $f = 1$ para o *hole* de mesmo índice em B^1 , e $f = 0$ para todos os demais *holes*.

Para cada *hole* de um *branch* só é atribuída uma única correspondência com *hole* de outro *branch*, de modo que os mesmos não podem ser usados para nenhuma outra correspondência entre *holes*. Os *holes* que não possuem interseção com *holes* do outro *branch*, ou cuja razão da interseção dividido pela união não foi a maior e, por isso, não foi estabelecido nenhum outro correspondente, são identificados como operações de *insert* ou *delete* de *hole*, conforme o caso. As operações de *insert* de *holes* se caracterizam pela existência de *inner ring* em B^1 sem correspondentes em B^A e tal fato é demonstração que naquele local, em B^A , não havia nenhum *hole* e que o mesmo foi inserido pelo usuário 1

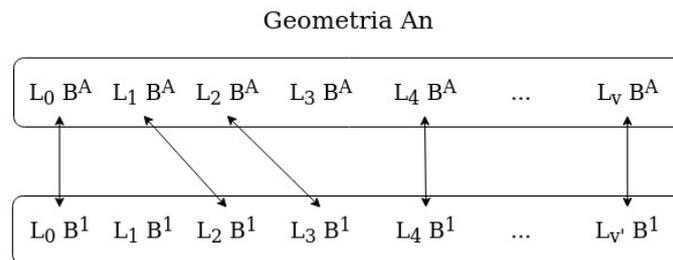
Tabela 5.1: Exemplo de tabela de pareamento de *holes*

		Inner Rings de B^1						
		$f_{tt'}$	H_1	H_2	H_3	H_4	...	$H_{v'}$
Inner Rings de B^A	H_1	f_{11}	f_{12}	0	0			0
	H_2	0	f_{22}	f_{23}	0			0
	H_3	0	0	0	0			0
	H_4	0	0	0	1			0
	...							
	H_v	0	0	0	0			1

com $1 > f_{12} > f_{11} > 0$, $1 > f_{23} > f_{22} > 0$, e $f_{12} > f_{22}$

e, por isso, é uma operação de *insert* de *hole*. De modo análogo, as operações de *delete* de *holes* se caracterizam pela existência de *inner ring* em B^A sem correspondentes em B^1 e tal fato é a demonstração que naquele local, em B^A , havia um *hole* e que o mesmo foi deletado pelo usuário 1 e, por isso, é uma operação de *delete* de *hole*.

Analisando os valores da Tabela 5.1, tem-se que $H_1 B^A$ tem como correspondente $H_2 B^1$, pois $f_{12} > f_{11}$ e também $f_{12} > f_{22}$; e que $H_2 B^A$ tem como correspondente $H_3 B^1$, pois $f_{23} > f_{22}$; e ainda que de $H_4 B^A$ em diante, tem como correspondente o *hole* de mesmo índice em B^1 . Já $H_3 B^A$ não tem correspondência com nenhum *hole* de B^1 e, como já dito, por existir em B^A e não mais existe em B^1 , é um *delete* de *hole*. Já $H_1 B^1$ tem uma correspondência com $H_1 B^A$, porém não é o maior valor. Por não ter nenhuma outra correspondência, acaba ficando sem correspondência em B^A . Dessa forma, por existir em B^1 e não existir em B^A , é um *insert* de *hole*. A Figura 5.12 apresenta as correlações entre os *holes* conforme analisado.

Figura 5.12: Exemplo de pareamento de *hole* em geometrias áreas

Após estabelecidas as correspondências entre os *inner rings* e também estabelecidas as operações de *insert* e *delete* de *hole*, é feito o mesmo processamento executado para as geometria do tipo linha. Individualmente, para cada conjunto de *rings*, incluindo o *ring* de delimitação externa da área (L_0), é feito o alinhamento de vértices, seguido da identificação dos segmentos alterados para cada conjunto de *rings* pareados entre B^A e B^1 . O exemplo 3, a ser apresentado nas próximas seções, retoma os conceitos apre-

sentados e permite reforçar o entendimento do alinhamento de vértices para os *rings* das áreas.

5.3.2 Aplicação de modificação complementar de geometrias do tipo área

A partir da proposição de modificação complementar da geometrias, é apresentada a sua aplicação para geometrias do tipo área. Da mesma forma que o estabelecimento de segmentos alterados deve ser feito por conjunto *rings* correspondentes, a definição de modificação complementar deve também ser feita por comparação de *rings* correspondentes. Dessa forma, com base nos *rings* presentes em B^A , são feitas as comparações entre B^A e B^1 e também entre B^A e B^2 . Analisando as operações feitas nos *rings* tem-se as seguintes situações possíveis:

1. B^1 realizou um *update* de segmento no *ring*, podendo ser por *insert* ou *delete* de vértice; e B^2 também realizou um *update* de segmento no *ring*, podendo ser por *insert* ou *delete* de vértice: atuar da mesma forma que para linhas, para estabelecer se há possibilidade de ser modificação complementar. Se atender aos critérios, olhando o *ring* como uma linha isolada, é possível integrar os *rings*, caso contrário haverá um conflito no *ring*;
2. B^1 (ou B^2) realizou um *update* de segmento no *ring*, podendo ser por *insert* ou *delete* de vértice; e outro *branch* deletou o *ring*: conflito;
3. B^1 (ou B^2) realizou um *update* de segmento no *ring*, podendo ser por *insert* ou *delete* de vértice; e outro *branch* não fez nenhuma alteração no *ring*: aplicar as alterações no *ring*;
4. B^1 deleta o *ring*; e B^2 também deleta o *ring*: nenhum conflito e o *ring* é removido da geometria resultante da integração;
5. B^1 (ou B^2) deleta o *ring*; e outro *branch* não fez nenhuma alteração no *ring*: o *ring* é removido da geometria resultante da integração;

Os casos em que há *insert* de *ring* devem ser tratados de maneira particular, pois os *rings* não existiam em B^A :

1. Se houver *insert* de *rings* por somente um dos *branches*, os mesmos devem ser inseridos na geometria resultante.

2. se houver *insert* de *rings* pelos dois *branches*, é necessário primeiro tentar fazer uma correlação entre os *rings* inseridos. As ações a serem executadas dependem do resultado da correlação:

- Aqueles que não tiverem correlação, devem ser inseridos na geometria resultante.
- Aqueles que tiverem correlação com o *ring* do outro *branch* só devem ser adicionados a geometria resultante se os mesmos forem iguais. Caso contrário, é um conflito, pois houve um *insert* de *hole* em uma mesma região, porém com variação na linha definição do *hole*.

5.3.3 Exemplo - Caso 3

Nesta seção, é apresentado o uso de modificação complementar da geometrias para o Caso 3, com base nas seguintes versões da geometria A1:

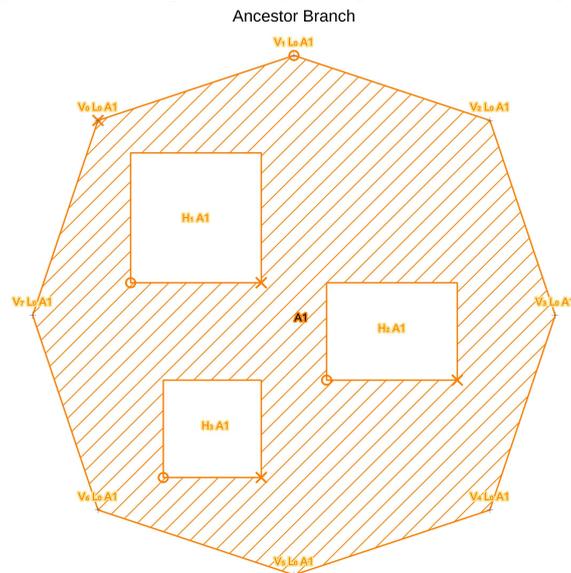
Ancestor Branch - A1: POLYGON ((252 398,255 399,258 398,259 395,258 392,255 391,252 392,251 395,252 398), (254.5 395.5,252.5 395.5,252.5 397.5,254.5 397.5,254.5 395.5), (257.5 394,255.5 394,255.5 395.5,257.5 395.5,257.5 394), (254.5 392.5,253 392.5,253 394,254.5 394,254.5 392.5))

Branch Usuário 1 - A1: POLYGON ((252 398,258 398,259 395,258 392,252 392,251 395,252 398), (257.5 394,255.5 394,254 397.5,257.5 397.5,257.5 394), (254.5 392.5,253 392.5,253 394,254.5 394,254.5 392.5), (256 393.5,257 393.5,257 392.5,256 392.5,256 393.5))

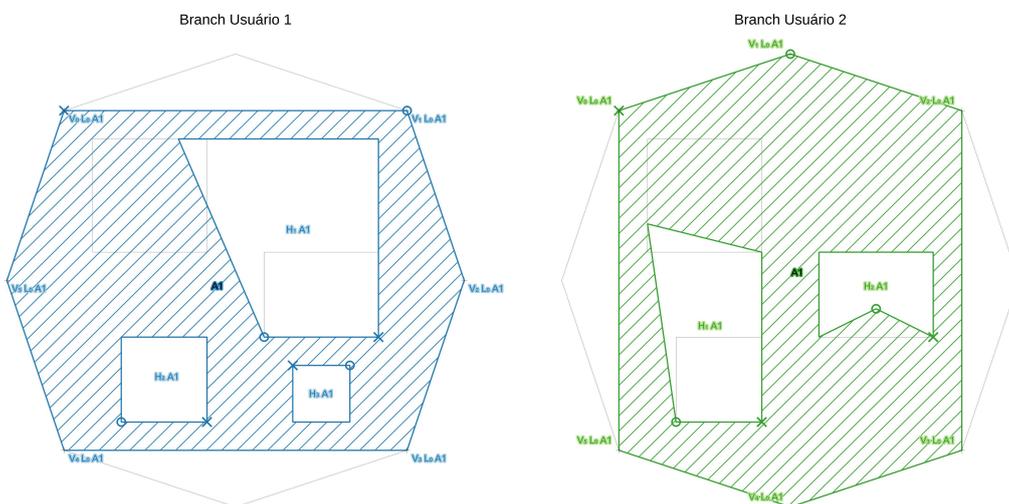
Branch Usuário 2 - A1: POLYGON ((252 398,255 399,258 398,258 392,255 391,252 392,252 398), (254.5 392.5,253 392.5,252.5 396,254.5 395.5,254.5 392.5), (257.5 394,256.5 394.5,255.5 394,255.5 395.5,257.5 395.5,257.5 394))

A representação espacial das versões da geometrias são apresentadas na Figura 5.13.

Figura 5.13: Representação espacial do Exemplo 3



(a) *Ancestor Branch*



(b) *Branch Usuário 1*

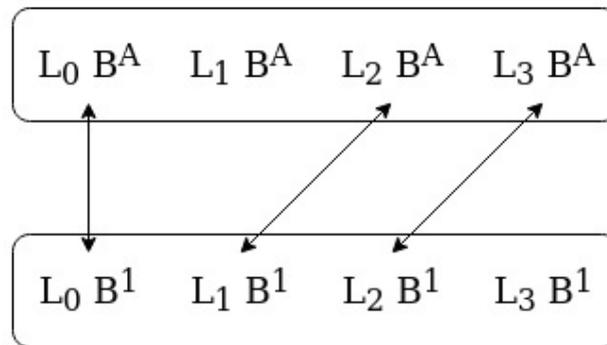
(c) *Branch Usuário 2*

Fazendo a análise de pareamento de *holes*, entre B^A e B^1 , são identificados os valores constantes da Tabela 5.2. A representação da correspondência dos *holes* entre versões B^A e B^1 da geometria A1 é apresentada na Figura 5.14.

Tabela 5.2: Exemplo 3 - tabela de pareamento de *holes* entre B^A e B^1

Inner Rings de B^A	Inner Rings de B^1			
	$f_{tt'}$	H_1	H_2	H_3
H_1	0.02	0	0	
H_2	0.31	0	0	
H_3	0	1	0	

Figura 5.14: Pareamento de *holes* das versões B^A e B^1 da geometria A1 do Exemplo 3



Fazendo a análise de *insert* e *delete* de *holes* e a análise de modificações entre B^A e B^1 , por *rings* pareados, tem-se:

- $L_0 B^A$ e $L_0 B^1$:
 - $V_1 L_0 B^A$: *delete* do respectivo vértice em B^1 ;
 - $V_5 L_0 B^A$: *delete* do respectivo vértice em B^1 ;
- $L_1 B^A$: por não ser o maior valor $f_{tt'}$ para $H_1 B^1$ e por não ter nenhum outro valor de $f \neq 0$, $L_1 B^A$ não tem correspondente em B^1 . Essa configuração indica que $H_1 B^A$ foi deletado;
- $L_2 B^A$ e $L_1 B^1$:
 - $V_2 L_2 B^A$ e $V_3 L_2 B^A$: *delete* dos respectivos vértices de B^A , com o *insert* de $V_2 L_1 B^1$ e $V_3 L_1 B^1$, promovendo uma movimentação dos vértices;
- $L_3 B^A$ e $L_2 B^1$:
 - não há alterações entre as versões;

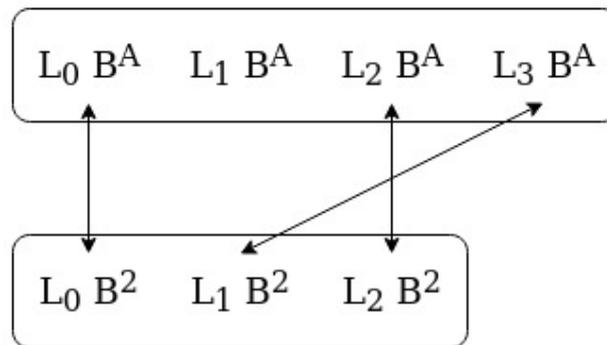
- $L_3 B^1$: não há nenhum valor $f_{tt'}$ para $H_t B^A$, que $f \neq 0$. Dessa forma, $L_3 B^1$ não tem correspondente em B^A . E essa configuração indica que $H_3 B^1$ foi inserido na geometria.

Fazendo a análise de pareamento de *holes*, entre B^A e B^2 , são identificados os valores constantes da Tabela 5.3. A representação da correspondência dos *holes* entre versões B^A e B^2 da geometria A1 é apresentada na Figura 5.15.

Tabela 5.3: Exemplo 3 - tabela de pareamento de *holes* entre B^A e B^2

Inner Rings de B^A	Inner Rings de B^2		
	$f_{tt'}$	H^1	H^2
H^1		0.05	0
H^2		0	0.8
H^3		0.4	0

Figura 5.15: Pareamento de *holes* das versões B^A e B^2 da geometria A1 do Exemplo 3



Fazendo a análise de *insert* e *delete* de *holes* e a análise de modificações entre B^A e B^2 , por *rings* pareados, tem-se:

- $L_0 B^A$ e $L_0 B^2$:
 - $V_3 L_0 B^A$: *delete* do respectivo vértice em B^2 ;
 - $V_7 L_0 B^A$: *delete* do respectivo vértice em B^2 ;
- $L_1 B^A$: por não ser o maior valor $f_{tt'}$ para $H_1 B^2$ e por não ter nenhum outro valor de $f \neq 0$, $L_1 B^A$ não tem correspondente em B^2 . Essa configuração indica que $H_1 B^A$ foi deletado;
- $L_2 B^A$ e $L_1 B^2$:
 - $V_1 L_1 B^2$: *insert* do respectivo vértice em B^A , entre $V_0 L_2 B^A$ e $V_1 L_2 B^A$;

- $L_3 B^A$ e $L_1 B^2$:

- $V_2 L_3 B^A$ e $V_3 L_3 B^A$: *delete* dos respectivos vértices de B^A , com o *insert* de $V_2 L_1 B^A$ e $V_3 L_1 B^A$, promovendo uma movimentação dos vértices;

Comparando as alterações entre B^A-B^1 e B^A-B^2 , com base nos *rings* pareados, tem-se a seguinte situação:

- *rings* $L_0 B^A - L_0 B^1$ e $L_0 B^A - L_0 B^2$:

1. $V_0 L_0 B^A$ a $V_2 L_0 B^A$: tem-se o *delete* de $V_1 L_0 B^A$, por parte de B^1 ;
2. $V_2 L_0 B^A$ a $V_4 L_0 B^A$: tem-se o *delete* de $V_3 L_0 B^A$, por parte de B^2 ;
3. $V_4 L_0 B^A$ a $V_6 L_0 B^A$: tem-se o *delete* de $V_5 L_0 B^A$, por parte de B^1 ;
4. $V_6 L_0 B^A$ a $V_8 L_0 B^A$: tem-se o *delete* de $V_7 L_0 B^A$, por parte de B^2 ;

- *rings* $L_1 B^A - B^1$ e $L_1 B^A - B^2$:

1. $L_1 B^A$: tem-se o *delete* de $H_1 B^A$, tanto por parte de B^1 , por parte de B^2 ;

- *rings* $L_2 B^A - L_1 B^1$ e $L_2 B^A - L_2 B^2$:

1. $V_0 L_2 B^A$ a $V_1 L_2 B^A$: tem-se o *insert* de $V_1 L_2 B^2$, por parte de B^2 ;
2. $V_1 L_2 B^A$ a $V_5 L_2 B^A$: tem-se a movimentação dos vértices $V_2 L_2 B^A$ e $V_3 L_2 B^A$ para as posições $V_2 L_1 B^1$ e $V_3 L_1 B^1$;

- *rings* $L_3 B^A - L_2 B^1$ e $L_3 B^A - L_1 B^2$:

1. sem alteração por parte de B^1 ;
2. $V_1 L_3 B^A$ a $V_5 L_3 B^A$: tem-se a movimentação dos vértices $V_2 L_3 B^A$ e $V_3 L_3 B^A$ para as posições $V_2 L_1 B^2$ e $V_3 L_1 B^2$;

- *rings* $L_3 B^1$:

1. $L_3 B^1$: tem-se o *insert* de $H_3 B^1$, sem correspondentes tanto em B^A , como em B^2 ;

Aplicando os critérios de modificação complementar de geometrias, tem-se:

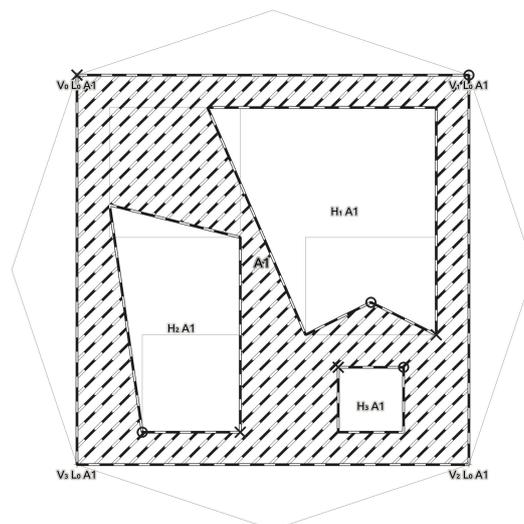
- $L_0 B^A$: as alterações são feitas em segmentos distintos e, portanto, é possível integrar o *ring*;

- $L_1 B^A$: mesma operação de *delete* promovida por ambos usuários, o *ring* é removido da versão final da geometria.
- $L_2 B^A$: as alterações são feitas em segmentos distintos e, portanto, é possível integrar o *ring*;
- $L_3 B^A$: as alterações são feitas por somente um usuário, portanto, é possível integrar o *ring*;
- $L_3 B^1$: *insert de ring* promovido por um usuário, sem correspondente na outra versão, o *ring* é inserido na versão final da geometria;

De acordo com as análises realizadas, é possível integrar as duas versões, B^1 e B^2 , sendo a geometria final expressa pelo WKT: POLYGON ((252 398,258 398,258 392,252 392,252 398), (257.5 394,256.5 394.5,255.5 394,254 397.5,257.5 397.5,257.5 394), (254.5 392.5,253 392.5,252.5 396,254.5 395.5,254.5 392.5), (256 393.5,257 393.5,257 392.5,256 392.5,256 393.5)) e a sua representação espacial expressa na Figura 5.16.

Figura 5.16: Geometria A1 integrada

Merged Branch - Geometria integrada



5.3.4 Operações do tipo *flip* e rotação de índices em geometrias do tipo área

Para os *rings*, alterações do tipo *flip* e rotação de índice não são alterações que tenham correspondência com o mundo físico e só ocorrem pela forma de implementação computacional da definição do dado geográfico, seja ele WKT, WKB ou outro. Então essas alterações podem ser desconsideradas durante a definição dos conflitos.

5.3.4.1 Identificando alteração do tipo *flip* e seu tratamento durante a modificação complementar de geometrias

Diferente da forma que ocorre em geometrias do tipo linha, para geometrias do tipo área, somente a combinação de alterações pode ocasionar uma alteração do tipo *flip* nos *rings*. Para identificar as alteração do tipo *flip*, após feito o pareamento entre os *rings*, é feito o mesmo procedimento de identificação de *flip* que é feita para as linhas, para cada conjunto de *rings*. O seu tratamento durante a modificação complementar de geometrias é simplesmente desconsiderar essa alteração. Para desconsiderar as alterações do tipo *flip* de *ring* basta considerar todas as classes de área como sendo de tratamento especial como definido na Subseção 5.2.5. Estes procedimentos permitem o tratamento correto de alterações do tipo *flip* para as áreas.

5.3.4.2 Identificando alteração do tipo rotação de índices e seu tratamento durante a modificação complementar de geometrias

A semelhança da alteração de *flip* em *rings*, o resultado de várias operações pode ocasionar uma alteração do tipo rotação de índices nos *rings* das áreas. A identificação dessa alteração é feita após o pareamento dos *rings*, comparando o índice do primeiro vértice igual entre as versões a serem alinhadas. Se o vértice não tiver o mesmo índice, existe uma rotação que deve ser corrigida. Para promover a correção da rotação, a geometria posterior é modificada, promovendo uma nova rotação, de modo que o primeiro vértice comum nas duas geometrias tenha o mesmo índice. Tal modificação como já dito, não altera em nada as propriedades físicas do *ring*, apenas altera sua definição em WKT. Estes procedimentos devem ser feitos após a correção da alteração de *flip* e permitem o tratamento correto de alterações do tipo rotação de índice para as áreas.

5.4 Considerações finais sobre a proposta

Como condição adicional para fins de execução e correto funcionamento do método de modificação complementar só serão aceitos como *input* geometrias que atendam os requisitos de validade da OGC-SFA. Da mesma forma, após realizar a integração das geometrias, as mesmas são verificadas com relação aos requisitos de validade da OGC-SFA e, caso falhem, são retornadas como conflito para resolução manual por parte dos usuários. O método proposto promove uma alteração do tipo de *Atomic Data Unit* do atributo geometria. Ao invés de usar o atributo como valor atômico, permite que se use a subparte da geometria, sendo o segmento da geometria dinamicamente definido de acordo com os dados sendo trabalhados. A partir do segmento é que são feitas as análises para permitir a integração da geometria, seu *merge*. Outro ponto diferencial a respeito da proposta é que a mesma permite a integração de geometrias que tenham sofrido alterações ainda mais profundas, como *flip* ou rotação de índice de *ring* ou alteração de ordem dos *holes*, fato não visto em nenhuma outra pesquisa ou sistema disponível para uso com dados geográficos. A proposta de modificação complementar de geometria possibilita a resolução de diversos potenciais conflitos dos sistemas tradicionais, particularmente das operações que envolvam *update* de geometrias.

A estratégia de integração usada pelo método descrito é baseada no modelo de funcionamento observados nos softwares de versionamento de dados geográficos, que é do tipo binária. Esse tipo de estratégia faz a integração de duas versões por vez. Ainda que o enfoque da solução tenha sido a estratégia binária, o modo como o método é desenvolvido permite sua expansão para uso em estratégia n-ária, desde que se use uma versão *ancestor* adequada a todas as versões que serão integradas, pois as comparações são feitas entre o *ancestor* e cada versão posterior. O uso da estratégia n-ária pode permitir menor quantidade de *merge* de versionamento, o que demanda menos tempo. Porém, pela possibilidade de ter mais alterações a serem integradas, esta estratégia tem a tendência a apresentar modificações mais complexas e que não atendam aos critérios estabelecidos, o que resultará em maior complexidade para resolução manual pelos usuários.

6 ESTUDO DE CASO - 1º CENTRO DE GEOINFORMAÇÃO

Este capítulo descreve o estudo de caso realizado no 1º Centro de Geoinformação (1º CGEO) do Exército Brasileiro com o objetivo de avaliar qualitativamente a proposta de Modificação Complementar de Geometria apresentada nesta dissertação de mestrado. Foi utilizada uma metodologia de pesquisa centrada em questionário com profissionais (especialistas) que atuam na coordenação da elaboração de base de dados geográficos do Exército Brasileiro.

A suposição avaliada é: *É possível reduzir a interação manual ao integrar dados de diferentes versões do mesmo banco de dados geográfico?*. Essa suposição foi elaborada com base no uso anterior de sistemas de versionamento onde sua utilização, apesar de potencialmente gerar benefícios, sempre foi acompanhada por um grande esforço de trabalho na resolução de conflitos.

Este capítulo primeiramente descreve o protocolo e a metodologia utilizados. Em seguida, os resultados são detalhadamente descritos. Por fim, são apresentadas a análise geral dos resultados e as limitações do estudo de caso.

6.1 Metodologia

Para a realização deste estudo de caso, foi firmada uma parceria com a Divisão de Geoinformação, área responsável pela produção de dados geográficos do 1º Centro de Geoinformação. O 1º CGEO é uma das cinco unidades da Diretoria de Serviço Geográfico (DSG), do Exército Brasileiro, incumbida da construção de bases de dados e cartas topográficas, sendo responsável direta pela elaboração de dados e produtos da região sul do Brasil para atendimento de necessidades da sociedade brasileira e do próprio Exército. Esta diretoria é uma das responsáveis, junto com o IBGE, pela elaboração do mapeamento de base do Brasil. Em conjunto com o 1º CGEO foram elaborados os dados a serem usados provenientes do Projeto de Mapeamento da Região Funcional 1, do estado do Rio Grande do Sul, o qual foi elaborado na unidade militar.

Foi utilizada uma metodologia de pesquisa centrada em um questionário com profissionais do exército (especialistas) que atuam na coordenação da elaboração de base de dados geográficos. Estes usuários tinham como tarefa responder o questionário, fazendo

uso, quando necessário, dos *softwares QGIS*¹ e *Kart*², além de rotinas elaboradas para implementar a proposta de modificação complementar de geometrias e responder perguntas sobre suas percepções. As rotinas que implementam o método de modificação complementar foram desenvolvidas em python para uso com a API do software QGIS (pyqgis).

No total, três profissionais participaram do experimento como voluntários, sendo esta também a quantidade de pessoas aptas, selecionadas pela coordenação do experimento no 1º CGEO. Os convites foram restritos aos profissionais que já tinham familiaridade com a coordenação de produção de dados geográficos, que tivessem participado em atividades de confirmação em campo e que já tivessem familiaridade com sistemas de versionamento, preferencialmente versionamento de dados geográficos, de modo que pudessem ter alguma familiaridade com o tema. Cabe ressaltar que, devido a especificidade do tema de pesquisa e a necessidade de especialistas para participar do estudo de caso, o foco foi na análise qualitativa e argumentativa dos resultados e não na quantidade de participantes.

O efetivo selecionado foi convidado a executar um conjunto de tarefas e responder um questionário o qual se encontra no apêndice A. O questionário consiste de quatro partes:

1. informações pessoais (demografia);
2. familiarização com versionamento de dados geográficos e resolução de conflitos de geometria;
3. apresentação do conceito de Modificação Complementar de Geometria;
4. sequência de tarefas que são avaliadas pelas respostas do questionário.

A seção de familiarização com versionamento de dados geográficos e resolução de conflitos (item 2) e a seção de apresentação do conceito de Modificação Complementar de Geometria (item 3) foram introduzidas para estabelecer uma base comum aos voluntários especialistas, de modo que todos tivessem os mesmos conceitos para avaliar o método proposto. Neste estudo de caso, vamos apresentar a demografia e as tarefas executadas e analisadas pelos especialistas.

¹GIS Open Source já usado pelo 1º CGEO <<https://qgis.org/>>

²Sistema de versionamento dados geográficos <<https://kartproject.org/>>

6.2 Resultados

Esta seção descreve detalhadamente os resultados obtidos neste estudo de caso. Primeiramente, a demografia dos participantes é apresentada. Em seguida, cada tarefa executada pelos voluntários é descrita, sendo as respostas analisadas.

6.2.1 Demografia

Todos os voluntários são engenheiros cartógrafos, com mais de 10 anos de atuação na área de dados geográficos. E por desenho do experimento, foi restringido a pessoas que já participaram de atividades de confirmação em campo e atuaram na coordenação de produção de dados geográficos, pois com essa vivência já enfrentaram a necessidade de integração de bases oriundas de uma única fonte, bem como poderiam vislumbrar a aplicação do método desenvolvido, em ambiente de produção de dados. Outro fato ser destacado é o conhecimento anterior de sistemas de versionamento para que fosse possível vislumbrar a atuação em base única versionada e as complexidades em se atuar com versionamento de dados geográficos. Cabe ressaltar que pela especificidade do tema o experimento não contou com um universo de seleção amplo, sendo que o próprio universo foi integralmente voluntário e participou do experimento.

A baixa quantidade de dados impede análises estatísticas que generalizem mais as respostas, porém pelo fato dos voluntários serem especialistas na área, as respostas obtidas se traduzem num indicativo de relevância acerca do assunto.

6.2.2 Descrição das Tarefa e Análise dos Resultados

A presente subseção apresenta as tarefas que fizeram parte do questionário elaborado. Também traz os resultados esperados e uma breve análise do que foi respondido. Ao final de cada item são compiladas as respostas dos especialistas.

Após a primeira parte de coleta de informações pessoais dos voluntários, foi elaborada uma parte de familiarização com versionamento de dados geográficos e a resolução de conflitos de geometria, de modo a nivelar os conhecimentos dos participantes. Foi apresentada a Figura 6.1, com estado anterior comum, e os estados de duas versões paralelas que devem ser integradas e foi solicitado que respondessem: “Na sua opinião é possível

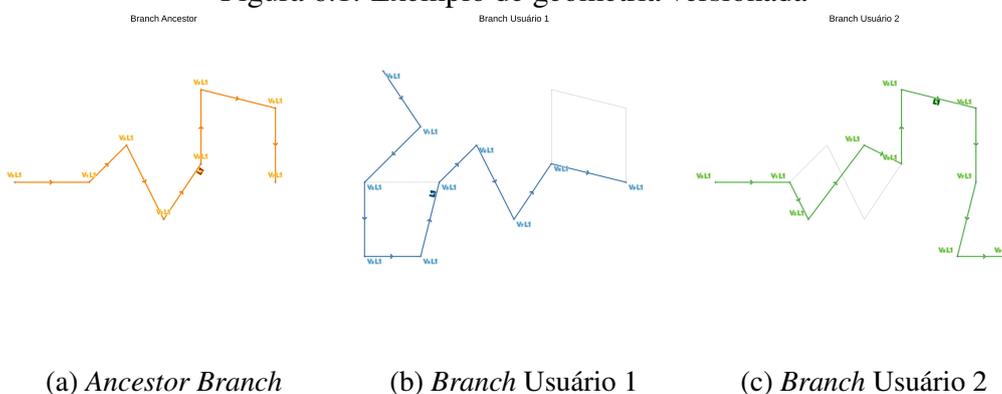
realizar o integração das geometrias?” A resposta esperada era o sentido que indicasse a viabilidade pelo fato das mudanças apresentadas serem espacialmente isoladas, sendo possível integrar as geometrias. Porém, dois dos especialistas indicaram o comportamento esperado de um sistema de versionamento de dados, no qual a mesma é um conflito. A seguir, encontram-se transcritas as respostas dos especialistas para o item:

Especialista 01: Sim, visto que as diferenças entre as feições são espacialmente isoladas o que permite a combinação das mesmas em uma única geometria.

Especialista 02: Sistemas de versionamento consideram que tais alterações são conflitos, já que a geometria foi alterada de forma distinta. Na prática uma das versões seria aceita como versão correta, e as alterações do outro usuário seria replicada manualmente.

Especialista 03: Não é possível, quando usamos um sistema que faz versionamento uma das alterações vai prevalecer, e as outras mudanças são posteriormente incorporadas de maneira manual.

Figura 6.1: Exemplo de geometria versionada



A seguir a tarefa envolvia o uso do *software Kart*, no qual foi inserida a mesma geometria da Figura 6.1 e foi perguntado “Ao realizar a integração das versões qual foi o resultado?”. O resultado esperado era a indicação de conflito de versionamento apresentado pelo *software* e todos os voluntários responderam dessa forma. A próxima pergunta, que tem relação com a anterior, foi “A respeito da questão anterior, qual sua opinião sobre o resultado e o possível impacto dele para o uso em um ambiente de produção de dados geográficos?”. O resposta esperada era no sentido de indicar que o resultado não favoreceria o uso do sistema de versionamento em produção de dados geográficos, pois o fato

desse tipo de configuração ser conflito para o versionamento resultava em mais trabalho na resolução de conflito, o que acarreta em maior tempo de produção. Todos os especialistas concordaram ao afirmar que o uso de um sistema de versionamento que apresentasse tal configuração de geometrias como conflito impacta de maneira negativa a produção de dados. A seguir, encontram-se transcritas as respostas dos especialistas para o item:

Especialista 01: Por ser o resultado um conflito, o mesmo não favorece o uso do sistema, já que as alterações são espacialmente separadas e seria possível realizar a integração das geometrias. O fato de um caso como apresentado ser reportado como conflito adiciona mais tempo no processo de produção, necessitando de interações manuais dos operadores o que pode introduzir mais erro também. Tal fato impacta o processo de produção de dados geográficos de maneira negativa.

Especialista 02: Em casos que é necessário considerar a alteração de ambos os usuários existe um retrabalho manual para garantir que ambas as mudanças estejam na versão integrada final. Esse trabalho manual implica em perda de tempo, e pode conter erros.

Especialista 03: Não será possível obter a geometria, pois há conflito. E por isso seu uso em produção traria impacto negativo.

Após a familiarização, foi introduzido o conceito de *Modificação Complementar de Geometria*, sendo apresentada a Figura 6.2, que mostra a integração das versões da geometria da Figura 6.1. Foi então perguntado “Na sua opinião, a integração está correta?”. A resposta esperada era no sentido de que a Figura 6.2 representa a aplicação dos conceitos de modificação complementar e, portanto, estaria correta. Todos os voluntários responderam que a imagem representa a aplicação correta. A seguir, encontram-se transcritas as respostas dos especialistas para o item:

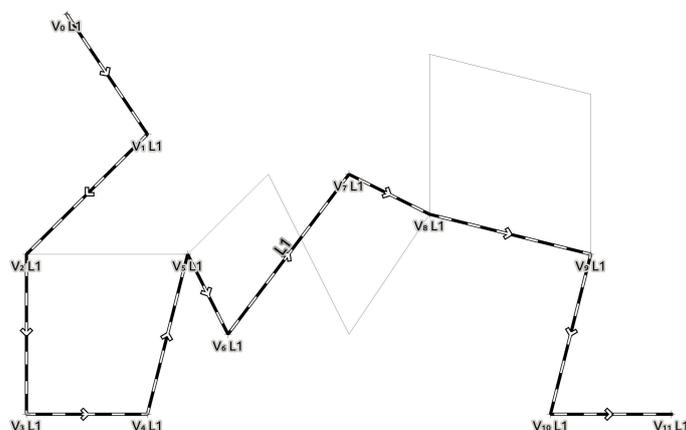
Especialista 01: Sim, pois a geometria final, após a integração consegue comportar as alterações de ambos os usuários do sistema, visto que as mesmas são espacialmente separadas.

Especialista 02: Sim, já que ambas as alterações feitas pelos usuários estão consideradas na geometria.

Especialista 03: Sim. A integração proposta na figura está correta.

Figura 6.2: Exemplo de geometria integrada

Branch Merged



A última parte do questionário apresentava as tarefas para teste do método de *Modificação Complementar de Geometria* em um conjunto de dados. A primeira tarefa se referia a utilização do *software Kart* para realizar a integração do conjunto de dados preparados para o teste, junto com a inspeção/análise dos dados/resultados no *software QGIS*. Foi perguntado “Ao realizar a integração da base de dados, qual foi o resultado?”. Era esperado que os voluntários informassem que muitas das geometrias não foram integradas gerando conflitos e, pela análise dos dados, seria possível ver que várias das geometrias possuem características de serem modificação que se complementam. Todos os voluntários responderam que foram identificados muitos conflitos, sendo que um deles apontou que houve conflito mesmo em modificações espacialmente separadas. A seguir, encontram-se transcritas as respostas dos especialistas para o item:

Especialista 01: Diversas feições foram consideradas como conflitos, mesmo que tenham modificações espacialmente separadas.

Especialista 02: Foram encontrados 78 conflitos.

Especialista 03: Muitos conflitos resultantes das geometrias alteradas por ambos os operadores.

A próxima tarefa solicitava a aplicação do método de “Modificação Complementar de Geometria”, o qual foi implementado por meio de *script* para uso no *software QGIS*.

Após a inspeção/análise dos dados/resultados foi perguntado a respeito do resultado obtido com o uso do método. Era esperado que os voluntários informassem a respeito da resolução de grande parte dos conflitos, já que parte das geometrias do conjunto de dados preparado atendia os critérios de modificação complementar. Após análise dos dados, os voluntários responderam que, de fato, houve redução no número de geometrias em conflito, e que as geometrias não resolvidas não atendia aos critérios estabelecidos de modificação complementar. A seguir, encontram-se transcritas as respostas dos especialistas para o item:

Especialista 01: As modificações que atendiam o critério de modificação complementar foram de fato integradas e não se apresentam mais como conflito.

Especialista 02: O número de conflitos reduziu para 23. Analisando os 23 casos são realmente discordâncias / erros entre os usuários.

Especialista 03: Foram resolvidos os conflitos que atendiam os critérios.

A última tarefa solicitava a análise dos resultados obtidos no item anterior e perguntava a respeito da opinião sobre o método de “Modificação Complementar de Geometria” proposto. Era esperado que o especialista indicassem que o método resolve em parte os problemas de integração de geometrias verificados com o uso dos sistemas de versionamento de dados geográficos hoje disponíveis. Os voluntários informaram que a proposta de fato traz ganho ao diminuir a quantidade de conflitos gerados durante a integração dos dados. Cabe ressaltar dois apontamentos: o ganho depende do conjunto de dados em trabalho; e a redução de erros que podem ser inseridos pela resolução manual dos conflitos. A seguir, encontram-se transcritas as respostas dos especialistas para o item:

Especialista 01: O método de modificação complementar conseguiu realizar a integração das feições com alterações espacialmente isoladas, conforme os critérios apresentados. Os conflitos remanescentes não atendem os critérios e por isso permanecem necessitando de resolução manual dos usuários. Apesar disso a solução se mostra muito eficiente pois consegue diminuir os casos de conflito, mesmo que a taxa de resolução tenha variação em função dos dados usados no sistema. O uso do método apresentado traz vantagens para a produção de dados geográficos, pois facilita a integração das versões e das geometrias contidas nelas.

Especialista 02: O método é aplicável na situação prática de produção de geoinformação no 1º Centro de Geoinformação, implicando na redução do trabalho manual para

considerar alterações em geometrias que abrangem múltiplas unidades de trabalho, como redes hidrográfica, rodoviária e ferroviária.

Especialista 03: O fato do método conseguir integrar as geometrias que antes se apresentavam em conflito é uma clara vantagem em relação a proposta anterior. Isso se traduz em mais agilidade para o processo de produção de dados, bem como uma redução na inserção de erros durante a resolução manual dos conflitos.

6.3 Análise Geral dos Resultados

Quando analisadas as respostas a cerca do conhecimento geral do assunto, os voluntários demonstraram que apesar de todos terem conhecimento de versionamento, pelo menos a respeito de DCVS, alguns puderam fazer análises mais aprofundadas sobre o comportamento esperado dos sistemas específicos de versionamento de dados geográficos, indicando um entendimento maior sobre a forma como o sistema trata os dados geográficos e sua determinação de conflitos.

Outro ponto de análise decorre do entendimento dos impactos possíveis em um ambiente de produção de dados geográficos. Todos os voluntários demonstraram que o fato do sistema apresentar como conflitos todas as situações em que houver mudança na geometria feita pelos dois usuários, mesmo que em regiões espacialmente isoladas, é uma desvantagem para o sistema, o que pode diminuir a sua adoção.

Porém, quando introduzido o conceito de Modificação Complementar de Geometria, e suas potencialidades de diminuição de conflitos, quando atendidos os critérios elencados, trazem a percepção de melhora na utilização do versionamento, bem como demonstra um ganho em relação a situação atual dos sistemas de versionamento. E conforme citado pelos voluntários, a diminuição de correção manual ajuda a diminuir também a possibilidade de introdução de erros durante o processo de produção de dados espaciais.

Conforme as respostas fornecidas pelos especialistas, com mais de 10 anos de trabalho em produção de dados espaciais, o método de modificação complementar de geometrias favorece a integração de geometrias em versionamento, diminuindo o conflito entre versões, reduzindo a possibilidade de erro e aumentando a velocidade pela diminuição do tempo de produção dos dados. Por essa razão, a suposição elaborada é verdadeira visto que o método facilita a integração dos dados oriundos de versões de uma mesma base de dados geográficos, proporcionando ainda ganhos em velocidade de produção e

diminuição de erros.

6.3.1 Limitações

O estudo de caso foi realizado com usuários produtores de base de dados geográficos, tendo em vista a falta de base de dados conhecidas e anotadas para avaliação.

Uma ameaça quanto a validade dos resultados é que o foram três voluntários que trabalham com o mesmo domínio de dados. Para trabalhos futuros, seria interessante replicar os experimentos em outros domínios de dados geográficos.

7 CONCLUSÃO

A presente dissertação apresenta uma proposta que visa modificar a integração de geometrias em sistemas de versionamento de dados geográficos. Apresentou como motivação o problema enfrentado por produtores de dados geográficos, que durante as atividades de elaboração de base de dados necessitam de apoio de sistemas de versionamento, mas que durante o uso do sistema, percebeu-se um aumento na demanda pela resolução de conflito de versionamento. Apresenta também a fundamentação e os trabalhos relacionados com esta dissertação. Traz também uma série de definições que visam auxiliar na elaboração da proposta. No tocante ao core do trabalho, é feita a conceituação de modificação complementar de geometrias e todo o detalhamento de sua aplicação em situação de versionamento do dados, inclusive em situações que alteram profundamente a geometria, como as alterações de *flip* e rotação de índice de *ring*. Apresenta também um estudo de caso para avaliação qualitativa da proposta.

No Capítulo 1 foi feita a introdução do trabalho bem como a motivação, com o apontamento do problema real encontrado durante o trabalho com versionamento de dados geográficos. Foi ainda estabelecido o objetivo geral desta dissertação que é especificar uma nova proposta para modificação complementar de geometrias em sistemas de versionamento de dados geoespaciais e foram ainda indicadas as contribuições do trabalho.

No Capítulo 2 foram introduzidos diversos conceitos que foram usados para embasar a proposta. Foram apresentados os tipos de dados que são usado para a construção de bases de dados vetoriais (Ponto, Linha e Área), bem como sua definição em WKT, padrão proposto pela OGC. Foi, ainda, apresentado o banco de dados espaciais que é um tipo especializado dos bancos de dados que permite o uso em aplicações envolvendo dados espaciais. Foram também apresentados diversos conceitos de versionamento, com destaque para suas partes; suas operações, a exemplo das operações *branch* e *merge*; e ainda conceitos como *Atomic Data Unit* e estratégias de concorrência. Também foram apresentadas algumas classificações como a baseada em local do repositório, onde existem os CVCS e DCVS, sendo os últimos os mais usados atualmente, com destaque para o *software* GIT.

No Capítulo 3, apresentaram-se os trabalhos relacionados com o tema. Foram apresentados trabalhos que promoviam o uso de SID, para uso em versionamento. Foram também apresentados trabalhos que promoviam alterações na modelagem, de modo a permitir maiores ganhos no gerenciamento de múltiplas versões. Foram, ainda, apre-

sentados trabalhos que focaram no melhoramento do algoritmo de *diff* que são usado em dados geográficos. Porém, nenhum deles apresentou contribuições para a integração de geometrias em versionamento. Foram apresentados alguns trabalhos relatando casos de uso, porém nenhum com premissas semelhante ao caso em estudo. E foram ainda apresentados alguns dos *softwares* que permitem o versionamento de dados geográficos, com destaque para o *GeoGig* e para o *Kart*. Nenhum dos *softwares* apresentados traz quaisquer refinamentos na integração de geometrias, apenas apontando o conflito quando as definições das geometrias não eram estritamente iguais.

No Capítulo 4 foram feitas as principais definições para auxiliar na elaboração da proposta. Primeiro, foram especificadas as operações que são feita em ambiente GIS, como as operações de *insert* e *delete* de feição, e as diversas operações de *update* de geometria, sendo todas elas apresentadas com ilustrações para melhor compreensão. Após foram introduzidos diversos tipos de interações em situação de versionamento de dados, de modo a melhor caracterizar o comportamento de comparação por igualdade estrita, sendo esse o mesmo comportamento dos sistemas atuais de versionamento. Essas definições servem para apoiar o entendimento de mudanças que podem ocorrer em dados geográficos e como tratar as mudanças com mais ênfase em sua parte geográfica.

O Capítulo 5 começa com a introdução do conceito de *Modificação Complementar de Geometria*, apresentando uma nova abordagem de modo a permitir o aproveitamento do trabalho dos diversos usuários dos sistemas de versionamento e reduzir a carga de trabalho na resolução de conflitos pelo não uso do potencial do dado geográfico. A seguir foi apresentada a forma de aplicar o conceito em geometrias do tipo linha, primeiro com a identificação dos segmentos alterados e depois com a consolidação das versões da geometria, quando as mesmas atendem os critérios de modificação complementar. Ainda no tocante às linhas é apresentado o caso especial do tratamento de operação do tipo *flip*, operação esta que promove uma mudança profunda, com inversão completa da geometria. Nesse ponto, a proposta demonstra a necessidade de se definir a priori qual o tipo de tratamento de *flip* deve ser aplicada a classe. A Seção de linhas é exemplificada tanto pela apresentação de um caso genérico como também com a apresentação de uma caso concreto onde ocorre a operação de *flip*. Após a apresentação dos conceitos para as geometrias linhas, é feita a abordagem para as geometrias áreas. Inicialmente, é feita a discussão da particularidade da possível existência de *holes* em áreas, e como fazer a correlação dos *holes* entre as versões, já que o ordenamento é um mero artifício previsto na definição do dado. Assim é definida a forma de promover o pareamento entre os *holes* das

diversas versões. Uma vez pareados, para cada conjunto de *rings* são feitos os mesmos tipos de tratamentos de mudanças especificados para as linhas, com a particularidade de tratamento *insert e delete de holes*, quando for o caso. De modo a exemplificar a proposição é apresentado um exemplo ilustrando as áreas. A seguir, são apresentados dois casos de alterações especiais possíveis para as áreas, que são a alteração do tipo *flip* e rotação de índice em *ring*, que, apesar de ocorrerem de forma involuntária, não representam uma mudança física na geometria. Essa característica permite que as mesmas sejam ignoradas e seu tratamento é apresentado ao final da seção de áreas.

A proposta de *Modificação Complementar de Geometria* permite o tratamento e a resolução de possíveis conflitos, pela integração das geometrias. Esta integração não foi identificada em nenhuma bibliografia consultada, bem como também não foi vista em nenhum dos sistemas disponíveis para versionamento de dados geográficos. A proposta permite que se promova um tratamento do dado geográfico de modo mais refinado, promovendo um ganho na definição do *Atomic Data Unit* da geometria. Os sistemas disponíveis promovem o tratamento da geometria como um todo e, em versionamento, usam somente a comparação por igualdade estrita. Esse fato traz como consequência a demanda de resolução de conflitos onde, pela natureza do dado geográfico, não haveria necessidade, já que ocorrem em situações que podem ser aceitas as alterações de ambas versões do dado. Ao refinar o *Atomic Data Unit* usado, saindo da geometria como um todo e passando a usar a subparte da geometria, com a definição dinâmica do segmento, a proposta agrega um melhor tratamento do dado geográfico. Cabe ressaltar que a proposta não comporta somente alterações simples, como acréscimo ou remoção de vértice(s), mas permite o tratamento de alterações profundas, como o *flip* de linhas, a alteração do ordenamento dos *holes* em geometrias áreas, ou ainda as que promovam rotação nos índices dos *rings*. O tratamento desse tipo de alterações garante robustez e maior ganho no uso da proposta.

A proposta foi avaliada por meio de questionário com especialistas. Tentou-se a busca por bases de dados reais de modo a permitir comparações padronizadas, porém não foi encontrada nenhuma base de dados que permitisse esse tipo de avaliação. Dessa forma, optou-se por avaliar aplicando questionários aos usuários. Devido a especificidade do tema e de forma a conseguir público que pudesse avaliar a proposta, foi feita uma parceria com o 1º Centro de Geoinformação (1º CGEO), Organização militar subordinada a Diretoria de Serviço Geográfico, do Exército Brasileiro. Esta diretoria é uma das responsáveis, junto com o IBGE, pela elaboração do mapeamento de base do Brasil. Em

conjunto com o 1º CGEO foram elaborados os dados a serem usados provenientes do Projeto de Mapeamento da Região Funcional 1, do estado do Rio Grande do Sul, o qual foi elaborado na unidade militar. Os participantes também foram selecionados dentre os quadros da unidade, com foco nos voluntários que tivessem experiência na fase de confirmação em campo, e também já atuaram na coordenação de projetos de mapeamento, de modo a ter conhecimentos que possibilitassem vislumbrar a aplicação da proposta. Foram selecionados três voluntários que preenchiam os requisitos, sendo esta a quantidade disponível. Todos os voluntários possuem mais de dez anos de experiência com dados geográficos. As respostas dos voluntários foram agrupadas e foi possível verificar que apesar de algumas divergências em relação ao correto funcionamento dos sistemas de versionamento de dados, todos os voluntários concordaram que o uso de comparação por igualdade estrita torna a adoção dos sistemas disponíveis pouco atrativa em vista do custo de resolução de conflitos. Após a apresentação da proposta de *Modificação Complementar de Geometria* todos os voluntários enfatizaram que o sistema atuou de forma coerente com os critérios definidos e que sua adoção permite um melhor aproveitamento de soluções de versionamento de dados já que diminui o trabalho manual de resolução de conflito e ainda permite ganhos de qualidade.

Por fim, cabe destacar que a proposta pode ser usada em conjunto com os sistemas de versionamento de dados geográficos hoje disponíveis, a exemplo do *Kart* e *GeoGig*, sendo possível sua aplicação como um pós-processamento dos conflitos determinados por essas ferramentas. Como definido no trabalho, o sistema não contempla as *multigeometrias* (*multipoint*, *multilinestring*, *multipolygon* e *geometrycollection*) já que esses tipos de geometrias comportam mais de uma parte geométrica dentro do mesmo identificador, o que, para o presente trabalho, representa um problema na identificação da relação correta entre as partes constituintes das geometrias entre as versões. Como sugestão de trabalhos futuros, ainda dentro do tema de versionamento de dados geoespaciais, é sugerido a expansão da solução para multigeometrias, a busca e identificação de redundância de informação inserida por usuários de *branches* distintos e deduplicação usando como uma de suas componentes a informação geográfica.

REFERÊNCIAS

- CHACON, S.; STRAUB, B. **Pro Git**. [S.l.]: Apress, 2014.
- CSERÉP, M.; GIACHETTA, R. Operation-based revision control for geospatial data sets. v. 12, 2015.
- DIRETORIA DE SERVIÇO GEOGRÁFICO. **ET-EDGV: Especificações Técnicas Para Estruturação de Dados Geoespaciais Vetoriais**. 3.0. ed. Brasília, 2017.
- ELMASRI, R. et al. **Sistemas de banco de dados**. [S.l.]: Pearson Addison Wesley São Paulo, 2005.
- FELUSAB, Y.; SREBROA, H.; TALA, Y. Gis versioning management—the approach of the survey of israel. **International Archives of Photogrammetry and Remote Sensing**, v. 38, 2010.
- GIT. **GIT: distributed is the new centralized**. 2005. <<https://git-scm.com/>>. Accessed: 2021-09-13.
- GÜTING, R. H. An introduction to spatial database systems. **the VLDB Journal**, Springer, v. 3, n. 4, p. 357–399, 1994.
- KHUDYAKOV, P. Y. et al. Version control system of cad documents and plc projects. In: IOP PUBLISHING. **Journal of Physics: Conference Series**. [S.l.], 2018. v. 1015, n. 4, p. 042020.
- KOC, A.; TANSEL, A. U. A survey of version control systems. **ICEME 2011**, 2011.
- KOORDINATES. **Kart: Distributed version-control for geospatial and tabular data**. 2021. <<https://kartproject.org/>>. Accessed: 2021-09-13.
- LE, H. H. et al. Database versioning and its implementation in geoscience information systems. **Computers & Geosciences**, Elsevier, v. 70, p. 44–54, 2014.
- LI, D. et al. A versioning method of vector spatial database. In: IEEE. **2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)**. [S.l.], 2010. v. 5, p. V5–558.
- LOCATIONTECH. **GeoGig: A Tool for Geospatial Data Management**. 2014. <<https://geogig.org/>>. Accessed: 2021-09-13.
- MEDEIROS, C. B.; JOMIER, G. Managing alternatives and data evolution in gis. In: **Proc. ACM/ISCA Workshop on Advances in Geographic Information Systems**. [S.l.: s.n.], 1993. p. 34–37.
- MEDEIROS, C. B.; JOMIER, G. Using versions in gis. In: SPRINGER. **International Conference on Database and Expert Systems Applications**. [S.l.], 1994. p. 465–474.
- OGC. **The Open Geospatial Consortium: Simple Feature Access - Part 1: Common Architecture**. [S.l.], 2011.
- SAMPAIO, T. V. M.; BRANDALIZE, M. C. B. **Cartografia geral, digital e temática**. [S.l.]: Programa de Pós-Graduação em Ciências Geodésicas, 2018.

SANFORD, J. **GitSpatial: A spatial API for your GitHub-hosted GeoJSON**. 2013. <<https://github.com/JasonSanford/gitspatial/>>. Accessed: 2021-09-13.

SHEKHAR, S. et al. Spatial databases-accomplishments and research needs. **IEEE transactions on knowledge and data engineering**, IEEE, v. 11, n. 1, p. 45–55, 1999.

SVEEN, A. F. Geomdiff—an algorithm for differential geospatial vector data comparison. **Open Geospatial Data, Software and Standards**, Springer, v. 5, n. 1, p. 1–11, 2020.

WADEMBERE, I.; OGAO, P. Geometry updating for geospatial data integration. **International Archives of Photogrammetry and Remote Sensing**, v. 38, p. 52–57, 2010.

APÊNDICE A — QUESTIONÁRIO DE AVALIAÇÃO DO MÉTODO

Questionário de avaliação do método “Modificações complementares de geometria” em sistemas de versionamento de dados geográficos

Este questionário faz parte de um projeto de mestrado do Instituto de Informática da Universidade Federal do Rio Grande do Sul (INF-UFRGS) e busca avaliar a proposta de “Modificação complementar de geometrias” em sistema de versionamento de dados geográficos. A proposta visa facilitar a integração de geometrias e proporcionar uma diminuição de geometrias em conflito durante a integração dos dados geográficos.

O tempo aproximado deste questionário é de 45 minutos e os resultados serão agrupados de forma anônima.

A.1 Informações pessoais

Esta seção inclui algumas perguntas pessoais para entender melhor o perfil dos entrevistados. Os dados individuais não serão divulgados.

1. Qual a sua formação?

Graduação: _____

Pós-graduação: _____

2. A quanto tempo atua com dados geográficos?

_____ anos

3. Tem conhecimento de sistema de versionamento? (Git, SVN, etc)

Sim [] Qual: _____

Não []

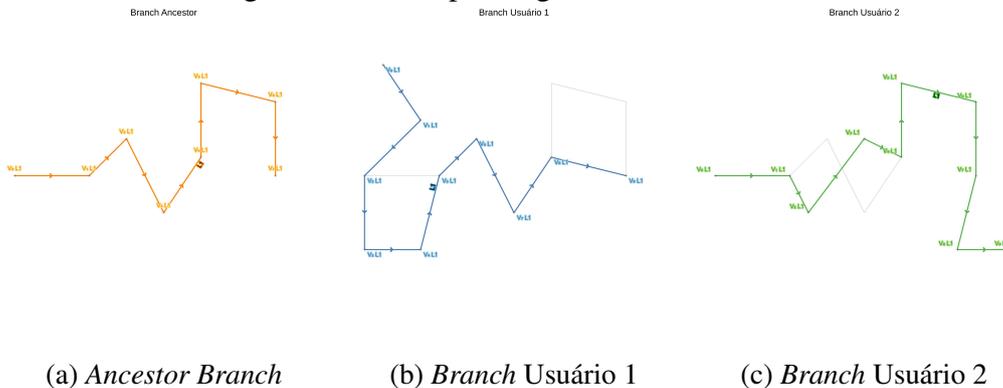
4. Já usou sistema de versionamento para dados geográficos? (Geogig, Kart, etc)

Sim [] Qual: _____

Não []

A.2 Familiarização com versionamento de dados geográficos e a resolução de conflitos de geometria

Figura A.1: Exemplo de geometria versionada



5. Na Figura A.1 são apresentados o estado anterior comum, e os estados de 2 versões paralelas que devem ser integradas. Na sua opinião é possível realizar o integração das geometrias?

Resposta:

6. A configuração da geometria apresentada na questão 5 está inserida no software de versionamento de dados geográficos *Kart*. Ao realizar o integração das versões qual foi o resultado?

Resposta:

7. A respeito da questão 6, qual sua opinião sobre o resultado e o possível impacto dele para o uso em um ambiente de produção de dados geográficos?

Resposta:

A.3 Conceito de modificação complementar de geometria

Modificação Complementar de Geometria acontece quando se atende aos seguintes critérios:

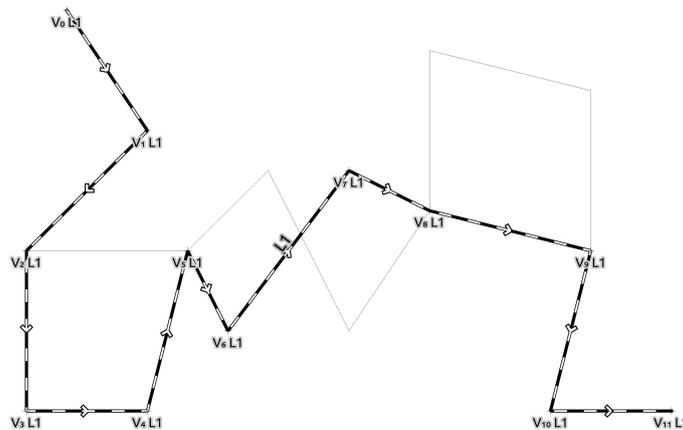
- Se um usuário modifica a geometria anterior, em um determinado segmento, o outro usuário não pode modificar esse mesmo segmento da geometria;

- Se os dois usuários modificam o mesmo segmento, as alterações devem ser iguais.

É admitido que segmento da geometria é qualquer subparte da geometria existente entre dois vértices, ou entre o início e um dos vértices, ou ainda entre um dos vértices e o fim da geometria, sendo necessário que os vértices que limitam os segmentos existam nas duas geometrias que serão comparadas.

Figura A.2: Exemplo de geometria integrada

Branch Merged



8. Com base na Figura A.1 é apresentada uma proposta de integração das geometrias seguindo a proposição de modificação complementar, presente na Figura A.2. Na sua opinião, a integração está correta?

Resposta:

A.4 Teste do método em um conjunto de dados

Um conjunto de dados foi preparado e inserido no software Kart para demonstração método, o qual deve ser usado para responder o próximo conjunto de questões.

9. Ao realizar a integração da base de dados, qual foi o resultado?

Resposta:

10. Ao aplicar o *script* que implementa método de “Modificação Complementar de Geometria”, qual foi o resultado?

Resposta:

11. Analisando os resultados obtidos na questão 10, qual sua opinião sobre o método de “Modificação Complementar de Geometria” proposto?

Resposta:

Obrigado por sua colaboração. Suas respostas ajudarão na avaliação do método proposto.