



**DETECÇÃO E RASTREAMENTO DE AGRUPAMENTOS  
DE INDIVÍDUOS MONITORADOS  
POR DISPOSITIVOS ELETRÔNICOS**

**PAULO LIMA MACHADO**

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**FACULDADE DE TECNOLOGIA**

**UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**DETECÇÃO E RASTREAMENTO DE AGRUPAMENTOS  
DE INDIVÍDUOS MONITORADOS  
POR DISPOSITIVOS ELETRÔNICOS**

**PAULO LIMA MACHADO**

**Orientador: PROF. DR. RAFAEL TIMÓTEO DE SOUSA JÚNIOR, ENE/UNB**

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA**

**PUBLICAÇÃO PPGENE.DM - 653/2016  
BRASÍLIA-DF, 29 DE NOVEMBRO DE 2016.**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**DETECÇÃO E RASTREAMENTO DE AGRUPAMENTO DE  
INDIVÍDUOS MONITORADOS POR DISPOSITIVOS ELETRÔNICOS**

**PAULO LIMA MACHADO**

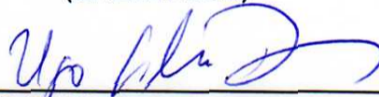
DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:



---

**RAFAEL TIMÓTEO DE SOUSA JR, Dr., ENE/UNB  
(ORIENTADOR)**



---

**UGO SILVA DIAS, Dr., ENE/UNB  
(EXAMINADOR INTERNO)**



---

**DANIEL BENEVIDES DA COSTA, Dr., UFC  
(EXAMINADOR EXTERNO)**

Brasília, 29 de novembro de 2016.

## **FICHA CATALOGRÁFICA**

PAULO LIMA MACHADO

**DETECÇÃO E RASTREAMENTO DE AGRUPAMENTOS DE INDIVÍDUOS MONITORADOS POR DISPOSITIVOS ELETRÔNICOS**

**2016xv, 74p., 201x297 mm**

(ENE/FT/UnB, Mestre, Engenharia Elétrica, 2016)

Dissertação de Mestrado - Universidade de Brasília

Faculdade de Tecnologia - Departamento de Engenharia Elétrica

## **REFERÊNCIA BIBLIOGRÁFICA**

PAULO LIMA MACHADO (2016) DETECÇÃO E RASTREAMENTO DE AGRUPAMENTOS DE INDIVÍDUOS MONITORADOS POR DISPOSITIVOS ELETRÔNICOS. Dissertação de Mestrado em Engenharia Elétrica, Publicação 653/2016, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 74p.

## **CESSÃO DE DIREITOS**

AUTOR: Paulo Lima Machado

TÍTULO: DETECÇÃO E RASTREAMENTO DE AGRUPAMENTOS DE INDIVÍDUOS MONITORADOS POR DISPOSITIVOS ELETRÔNICOS.

GRAU: Mestre ANO: 2016

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor se reserva a outros direitos de publicação e nenhuma parte desta dissertação de Mestrado pode ser reproduzida sem a autorização por escrito do autor.

---

Paulo Lima Machado

Quadra 17 Conjunto C Casa 17 - Sobradinho - Brasília - DF

## **AGRADECIMENTOS**

A minha família pela paciência e suporte nos momentos de dificuldade.

Ao meu orientador, Prof. Dr. Rafael Timóteo de Sousa Júnior, pelos conselhos e orientação, sendo fonte de inspiração nas horas mais complicadas.

Ao meu coorientador, Prof. Dr. Robson de Oliveira Albuquerque, que foi determinante no apoio para conclusão deste trabalho. Pela paciência em ouvir meus lamentos e pelos conselhos, sem os quais este trabalho não seria possível.

À Defensoria Pública da União – DPU, pela minha participação no projeto sobre o qual pude aprender e crescer como profissional.

Ao inestimável amigo Daniel Alves da Silva, por me incentivar a ingressar nesta jornada e por me apoiar incondicionalmente em todas as fases.

Aos colegas Guilherme de Lima Bernardes, por apoio na codificação da rotina em linguagem C e Hadoop, e Valério Martins pelo suporte prestado.

## **RESUMO**

### **DETECÇÃO E RASTREAMENTO DE AGRUPAMENTOS DE INDIVÍDUOS MONITORADOS POR DISPOSITIVOS ELETRÔNICOS**

**Autor: Paulo Lima Machado**

**Orientador: Professor Doutor Rafael Timóteo de Sousa Júnior**

**Programa de Pós-graduação em Engenharia Elétrica**

**Brasília, 29 de novembro de 2016**

A utilização de meios alternativos de vigilância de sentenciados, como o monitoramento via satélite, tem-se mostrado uma alternativa viável para fiscalização da execução de penas na forma de controle de indivíduos que infringem normas penais.

Tais sistemas de monitoramento utilizam as coordenadas geográficas (pontos) enviadas periodicamente pelas tornozeleiras e as compara com as zonas de inclusão e exclusão representadas por polígonos geográficos previamente cadastrados. Em caso de violação de perímetro, alertas são devidamente enviados às autoridades competentes.

A aplicação do sistema de monitoramento ocorre de forma reativa, pois as ações estão relacionadas a infrações que já ocorreram, restando à autoridades determinar apreensões ou investigações muitas vezes tardias.

Ao verificar os dados produzidos continuamente, há a possibilidade real de processá-los e gerar informações relevantes para auxiliar sistemas ou equipes a antever ou prevenir atos criminosos ainda em fase de elaboração e planejamento mudando o paradigma atual introduzindo ações proativas. Estas informações, se concentram inicialmente na identificação de encontros regulares (reuniões) de alguns indivíduos monitorados através da detecção de agrupamentos.

Neste contexto, este trabalho tem por objetivo definir, implantar e avaliar algoritmos de detecção de agrupamentos de indivíduos monitorados rastreando sua frequência, duração, elementos e localização. Tais algoritmos devem atender restrições inerentes ao sistema como limite de tempo de processamento. As informações geradas acerca dos agrupamentos de indivíduos apoiarão equipes técnicas de inteligência e investigação além de sistemas de análise de risco e apoio a decisão.

## **ABSTRACT**

### **DETECTION AND TRACEABILITY OF GROUPS OF INDIVIDUALS MONITORED BY ELECTRONIC DEVICES**

**Author: Paulo Lima Machado**

**Supervisor: Professor Rafael Timóteo de Sousa Júnior, PhD**

**Post-Graduation Program on Electrical Engineering**

**Brasilia, November 29th, 2016**

The use of alternative ways of sentencing surveillance, such as satellite monitoring, has been proved to be a viable alternative for enforcement of penalties in the form of control of individuals on parole.

Such monitoring systems use the geographical coordinates (points) sent periodically by the anklets and compare them with the inclusion and exclusion zones represented by geographic polygons previously registered. In case of perimeter violation, alerts are duly sent to the competent authorities.

The application of the monitoring system occurs in a reactive way, since the actions are related to infractions that have already occurred, leaving the authorities to determine seizures or investigations that are often late.

By verifying the continuously produced data, there is a real possibility of processing them and generating relevant information to assist systems or teams in anticipating or preventing criminal acts still in the planning and design stage by changing the current paradigm by introducing proactive actions. This information initially focuses on identifying regular meetings of some individuals monitored by clustering detection.

In this context, this work aims to define, deploy and evaluate algorithms for the detection of monitored individuals clusters by tracking their frequency, duration, location and number of individuals. Such algorithms must meet inherent constraints on the system as a time limit for processing. The information generated about the individuals clustering will support technical teams of intelligence and investigation in addition to systems risk analysis and decision support.

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	MOTIVAÇÃO	3
1.2	OBJETIVOS DO TRABALHO	4
1.3	METODOLOGIA DE PESQUISA	4
1.4	CONTRIBUIÇÃO DO TRABALHO	5
1.5	ORGANIZAÇÃO DO TRABALHO	5
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>7</b>
2.1	GEOPROCESSAMENTO	7
2.1.1	SENSORIAMENTO REMOTO	10
2.1.2	SISTEMA GLOBAL DE POSICIONAMENTO ( <i>Global Positioning System – GPS</i> )	10
2.1.3	IDENTIFICADOR DE REFERÊNCIA ESPACIAL ( <i>Spatial Reference Identifier – SRID</i> )	10
2.2	BANCOS DE DADOS GEOGRÁFICOS	11
2.2.1	POSTGIS SOBRE POSTGRESQL	11
2.3	COMPUTAÇÃO PARALELA	13
2.3.1	CLASSIFICAÇÃO DA COMPUTAÇÃO PARALELA	15
2.3.2	ESCALONAMENTO	17
2.3.3	MEMÓRIA COMPARTILHADA E MEMÓRIA DISTRIBUÍDA	18
2.3.4	MULTIPROCESSADORES SIMÉTRICOS ( <i>Symmetric MultiProcessors – SMP</i> )	18
2.4	COMPUTAÇÃO DISTRIBUÍDA	19
2.4.1	PROCESSADORES MASSIVAMENTE PARALELOS ( <i>Massively Parallel Processors – MPP</i> )	19
2.4.2	GRUPO DE COMPUTADORES ( <i>Clusters</i> )	19
2.4.3	COMPUTAÇÃO EM GRADE ( <i>Grid Computing</i> )	19
2.4.4	GEOPROCESSAMENTO PARALELO	20
2.5	REDES DE SENSORES DISTRIBUÍDOS SEM FIO ( <i>Wireless Sensor Network - WSN</i> )	22
2.5.1	FATORES QUE INFLUENCIAM O PROJETO DE REDES DE SENSORES	23
2.6	FUSÃO DE DADOS ( <i>Data Fusion</i> )	24



2.6.1	CLASSIFICAÇÃO DE FUSÃO DE DADOS .....	24
2.6.2	TÉCNICAS DE ASSOCIAÇÃO DE DADOS .....	25
2.6.3	CONSIDERAÇÕES ACERCA DA FUSÃO DE DADOS E O TRABALHO PRO- POSTO .....	26
<b>3</b>	<b>ESTADO DA ARTE .....</b>	<b>27</b>
3.1	PARTICIONAMENTO SIMPLES PARA DECOMPOSIÇÃO DE DOMÍNIOS ESPACIAIS .....	31
3.1.1	SUBDIVISÕES RECURSIVAS PARA DECOMPOSIÇÃO DE DOMÍNIOS ES- PACIAIS .....	31
3.2	SOBREPOSIÇÃO ESPACIAL VERSUS DECOMPOSIÇÕES SEM SOBREPOSIÇÃO.....	32
<b>4</b>	<b>DEFINIÇÃO DO PROBLEMA E PROPOSTA DE SOLUÇÃO.....</b>	<b>34</b>
4.1	PROBLEMA.....	34
4.1.1	DEFINIÇÃO DE PROXIMIDADE.....	35
4.1.2	DURAÇÃO MÍNIMA DE ENCONTROS PARA DEFINIÇÃO DE REUNIÃO .....	36
4.1.3	EXECUÇÃO DOS ALGORITMOS DENTRO DO TEMPO DEFINIDO .....	36
4.1.4	DIVISÃO DO ESPAÇO EM SUBÁREAS PARA PERMITIR PARALELIZAÇÃO .	37
4.1.5	SOBREPOSIÇÃO ESPACIAL PARA RESOLVER DEPENDÊNCIAS .....	38
4.1.6	IDENTIFICAÇÃO DO MESMO AGRUPAMENTO EM PROCESSAMENTOS SUBSEQUENTES .....	38
4.1.7	QUANTIDADE DE ELEMENTOS NO AGRUPAMENTO DETECTADO.....	38
4.1.8	FREQUÊNCIA DE ENCONTRO DOS AGRUPAMENTOS .....	39
4.1.9	HORÁRIOS COMUNS AOS ENCONTROS .....	39
4.1.10	ÁREA GEOGRÁFICA COMUNS AOS ENCONTROS.....	39
4.2	MODELO SISTÊMICO E PROPOSTA DE SOLUÇÃO .....	39
4.2.1	ETAPA 1 - DETECÇÃO DE PARES.....	41
4.2.2	ALGORITMO 1.1 – DIVISÃO DO ESPAÇO GEOGRÁFICO EM SUBÁREAS ..	42
4.2.3	ALGORITMO 1.2 – DETECÇÃO DE PARES EM SUBÁREAS .....	44
4.2.4	ETAPA 2 - DETECÇÃO DE GRUPOS .....	46
4.2.5	ETAPA 3 - DETECÇÃO DE RISCOS .....	47
<b>5</b>	<b>RESULTADOS E ANÁLISE .....</b>	<b>53</b>
5.1	EXECUÇÃO DO ALGORITMO 1.1 - DIVISÃO DO ESPAÇO GEOGRÁFICO EM SUBÁREAS .....	54
5.2	EXECUÇÃO DO ALGORITMO 1.2 - DETECÇÃO DE PARES EM SUBÁREAS	55
5.3	EXECUÇÃO DO ALGORITMO 2 - DETECÇÃO DE GRUPOS .....	56
5.4	EXECUÇÃO DO ALGORITMO 3 - DETECÇÃO DE RISCOS .....	57
5.4.1	CÁLCULO DA FREQUÊNCIA DOS ENCONTROS .....	58
5.4.2	CÁLCULO DO HORÁRIO MAIS COMUM DOS ENCONTROS .....	59
5.4.3	CÁLCULO DA ÁREA GEOGRÁFICA COMUM DOS ENCONTROS .....	59

5.5	EXECUÇÃO DOS TESTES DE DESEMPENHO .....	60
5.5.1	CENÁRIO 1 – PROCESSAMENTO EM PL/PGSQL SEM PARALELISMO ....	60
5.5.2	CENÁRIO 2 - PROCESSAMENTO EM PL/PGSQL COM PARALELISMO ....	62
5.5.3	CENÁRIO 3 - ALGORITMO 1.2 EM LINGUAGEM C SEM PARALELISMO ..	64
5.5.4	CENÁRIO 4 - ALGORITMO 1.2 EM LINGUAGEM C COM PARALELISMO .	64
5.5.5	CENÁRIO 5 - PROCESSAMENTO DISTRIBUÍDO EM <i>Big Data</i> COM <i>Hadoop</i>	66
5.6	ANÁLISE DO RESULTADO.....	66
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>68</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>71</b>

# LISTA DE FIGURAS

1.1	Evolução da população carcerária brasileira – Adaptado: DEPEN - Departamento Penitenciário Nacional. ....	1
1.2	Sistema de monitoramento por tornoeleiras. ....	2
2.1	Diferenças entre resoluções matriciais. Fonte: [Câmara and Monteiro 2004]...	9
2.2	Comparação entre representação vetorial e matricial. Fonte: [Câmara and Monteiro 2004].....	9
2.3	Single Instruction, Multiple Data – SIMD.....	16
2.4	Multiple Instruction, Multiple Data – MIMD.....	16
2.5	Algoritmo executado em paralelo com 3 processadores e MIMD. ....	17
2.6	Framework de Fusão de Dados JDL. Fonte: [Steinberg et al. 1999].....	25
2.7	Modelo conceitual do processamento de associação de dados. Fonte: [Castanedo 2013].....	26
3.1	Particionamento de rede alternado. Fonte: [Ding et al. 1992].....	32
3.2	Bisecção Alternativa Recursiva Dinâmica. Fonte: [Ding et al. 1992].....	32
4.1	Distância mínima entre dois pontos para definição de proximidade. ....	35
4.2	Aumento do cálculo de distâncias em função do aumento da quantidade de pontos.....	37
4.3	Etapas do processamento. ....	40
4.4	Subdivisão da área total em subáreas - Fonte: adaptado de [Ding and Densham 1996] .....	42
4.5	Expansão da área para contemplar pontos próximos em áreas adjacentes – Fonte: adaptado de [Ding and Densham 1996] .....	43
4.6	Algoritmo 1.1 – Distribuição dos pontos em subáreas. ....	43
4.7	Detalhamento da função recursiva do algoritmo 1.1. ....	43
4.8	Algoritmo 1.2 - Detecção de Pares. ....	45
4.9	Estrutura de execução dos algoritmos 1 e 2. ....	45
4.10	Algoritmo 2 - Detecção de Grupos. ....	46
4.11	Duração do grupo em processamentos consecutivos.....	47
4.12	Algoritmo 3 - Detecção de Riscos. ....	48
4.13	Cálculo de horas comuns aos encontros.....	51
4.14	Área geográfica comum do encontro. ....	51

4.15	Área geográfica comum a todos os encontros. ....	52
5.1	Pontos da massa de dados para teste dos algoritmos 1.1 e 1.2.....	55
5.2	Distribuição dos pontos nas subáreas após aplicação do algoritmo 1.1. ....	57
5.3	Grupos detectados durante período de tempo.....	58
5.4	Tempo de execução por algoritmo sem paralelismo. ....	61
5.5	Tempo de execução por máximo de pontos por subárea. ....	62
5.6	Tempo de execução do algoritmo de detecção de pares em linguagem PL/pgSQL. ....	63
5.7	Tempo de execução por linguagem.....	64
5.8	Tempo de execução do algoritmo de detecção de pares em linguagem C.....	65
5.9	Menor tempo de processamento por cenário.....	67

# LISTA DE TABELAS

2.1	Classificação do problema espacial conforme a característica. Adaptado de [Ding et al. 1992].....	21
4.1	Pontos coletados – Entrada do algoritmo 1 (Detecção de Pares).....	41
4.2	Estrutura de saída de dados da Etapa 1 (Detecção de Pares).....	41
4.3	Detalhamento dos passos do algoritmo 1.1 - Distribuição de Pontos em Subáreas. ....	43
4.4	Detalhamento dos passos da função recursiva. ....	44
4.5	Estrutura de saída do algoritmo 1.2 (Detecção de Grupos).....	44
4.6	Detalhamento dos passos do algoritmo 1 (Detecção de Pares). ....	45
4.7	Detalhamento dos passos do algoritmo 2 (Detecção de Grupos).....	46
4.8	Estrutura de saída do algoritmo 2 (Detecção de Grupos). ....	47
4.9	Passos do algoritmo 3 (Detecção de Riscos).....	49
4.10	Saída do algoritmo 3 (Detecção de Riscos). ....	49
5.1	Equipamento utilizado nos testes. ....	54
5.2	Amostra de dados para teste dos algoritmos 1.1 e 1.2. ....	56
5.3	Pares de pontos próximos resultante do algoritmo 1.2. ....	56
5.4	Grupos detectados pelo algoritmo 2.....	56
5.5	Dados de riscos por grupo resultante do algoritmo 3. ....	58
5.6	Dados de entrada para consulta de frequência de encontros. ....	59
5.7	Máximo de pontos por subárea e tamanho das subáreas. ....	62

# Capítulo 1

## Introdução

A utilização de tornozeleiras eletrônicas em pessoas investigadas e condenadas tem sido adotado por alguns estados brasileiros para tentar reduzir o encarceramento em massa no país. Segundo dados do Ministério da Justiça de dezembro de 2013 [DEPEN 2016], o Brasil possui uma das maiores populações prisionais do mundo, com 574.027 detentos. Os dados publicados pelo Ministério da Justiça até a publicação deste trabalho estava consolidado até meados de 2014.

Ao analisar o crescimento da população carcerária de 2005 a 2013 verificamos um aumento de 52% durante o período. Ao mesmo tempo, o crescimento das unidades prisionais não acompanha a demanda por novas vagas, conforme Figura 1.1.

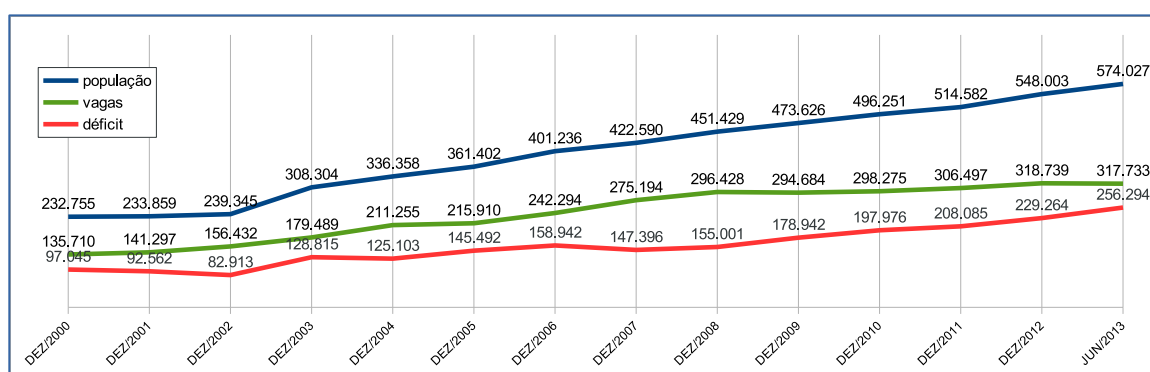


Figura 1.1: Evolução da população carcerária brasileira – Adaptado: DEPEN - Departamento Penitenciário Nacional.

A eficácia do cumprimento da pena dentro de estabelecimentos prisionais atuais tem sido questionada por doutrinadores afirmando que a prisão, como está sendo executada, fracassa como meio de reforma do indivíduo. A utilização de meios alternativos de vigilância dos sentenciados, no entanto, tem-se mostrado como uma opção viável para fiscalização da execução de penas, levando ao desenvolvimento de inovações na forma de controle de indivíduos que infringem normas penais. O monitoramento por satélite de detentos portadores de tornozeleiras eletrônicas tem mostrado-se como uma forma de vigilância efetiva nos Estados Unidos e na Europa [Barbosa 2010].

Sistemas de justiça criminal usam dispositivos GPS para monitorar infratores. Esses infratores são forçados por lei a usar tornozeleiras que informam suas localizações às agências de monitoramento [Daubal et al. 2013]. A Figura 1.2 ilustra uma das arquiteturas utilizadas em sistemas de monitoramento de sentenciados com tornozeleiras eletrônicas. Sistemas estes em que as coordenadas coletadas pelos dispositivos (1) são enviadas pela rede de transmissão de dados através de operadoras de telefonia celular (2) com destino à Central de Monitoramento (3). Esta última tem a função de processar os dados geográficos provenientes das tornozeleiras e gerar relatórios e alertas para que as autoridades competentes tomem providências de acordo com as políticas de cada sistema. A informação gerada mais comum é a indicação de violação de zonas proibidas.

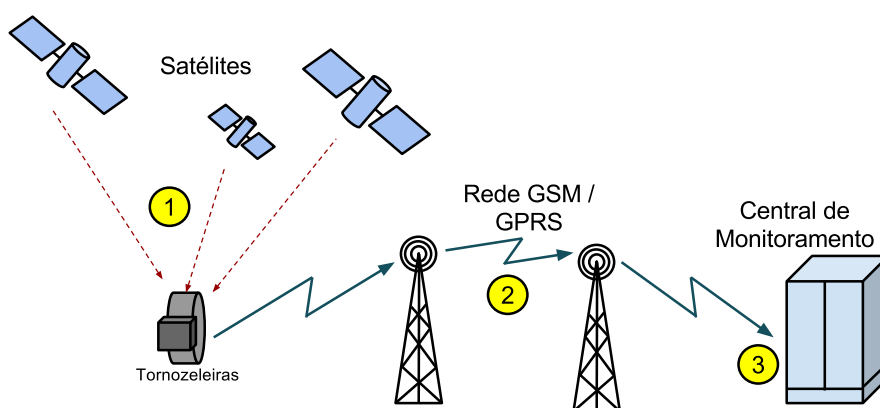


Figura 1.2: Sistema de monitoramento por tornozeleiras.

Os pacotes de dados transmitidos das tornozeleiras até a Central de Monitoramento possuem recursos de segurança como criptografia. Outros recursos relativos ao envio de alertas em caso de violação física do dispositivo também estão implementados. De toda forma, neste trabalho, consideramos que os pacotes são devidamente tratados e armazenados na Central de Monitoramento. E é sobre esta base de dados que o processamento de detecção de agrupamentos será aplicado.

Sistemas de telecomunicações de aplicação da Lei (*Law Enforcement Telecommunications Systems – LETS*) devem levar em conta o risco real representado por grupos específicos, considerando fatores como o nível de periculosidade de seus elementos e os tipos de infrações cometidas por cada um, entre outros. Assim, é relevante desenhar algoritmos capazes de fornecer dados para subsidiar a análise de risco e a tomada de decisão complementando este contexto.

Objetiva-se desse modo apoiar a geração de alertas informando prováveis formações de motins, preparação ou planejamento para ações criminosas, entre outras atividades suspeitas realizadas em grupos. Identificar antecipadamente os riscos mostra-se particularmente importante em épocas de grandes eventos sociais como, por exemplo, as Olimpíadas e a Copa do Mundo ocorridas recentemente no Brasil.

Considerando o contexto abordado acima, a importância da monitoração eletrônica é

ainda ressaltada em [Pimenta 2016]:

”Nesse sentido, é certo que a coleta e as diversas formas de tratamento de dados são atividades essenciais ao serviço de monitoração eletrônica. A gestão adequada das informações obtidas, inclusive quanto à geolocalização das pessoas monitoradas, é pressuposto para o bom funcionamento dos serviços, bem como para os processos de formulação, implementação, monitoramento e avaliação da política.”

Como solução tecnológica, o sistema de monitoramento por tornozeleiras pode ser visto pela ótica de Rede de Sensores Distribuídos Sem Fio (*Distributed Wireless Sensor Network*) [Lewis et al. 2004], que consiste na coleta de dados por dispositivos eletrônicos geograficamente distribuídos e envio destes dados através de uma rede. Neste contexto, aplica-se também algumas técnicas de Fusão de Dados abordado no Capítulo 2.

## 1.1 Motivação

O monitoramento eletrônico propicia economia para o sistema prisional, se levada em conta a questão do levantamento dos caracteres operacionais da vigilância. O provável sucesso do sistema de monitoramento de sentenciados poderá reduzir significativamente o número de processos graduativos de aquisição da liberdade atualmente utilizados, como a progressão de regime, saídas temporárias, dentre outras aplicações sem vigilância direta.

O uso do monitoramento eletrônico traz um grande avanço ao sistema de execução penal. A relevância para aplicação do monitoramento eletrônico é ainda reforçada pelos dados do relatório [Departamento Penitenciário Nacional - DEPEN 2014], onde:

- a. Do total de pessoas privadas de liberdade no Brasil, 41% estavam presas sem ainda terem sido julgadas (presos provisórios);
- b. Em números absolutos, o Brasil tem a quarta maior população de presos provisórios;
- c. A taxa de ocupação dos estabelecimentos prisionais brasileiros (161%) é a quinta maior, entre os países que contemplam o levantamento do DEPEN.

Tornar o sistema atual mais proativo pode aumentar significativamente o grau de confiabilidade no sistema judiciário brasileiro na medida que eventuais ações criminosas executadas em grupos podem ser desmontadas ainda na fase de planejamento.



## 1.2 Objetivos do Trabalho

O objetivo deste trabalho é a definição e implementação de algoritmos de processamento para detecção e acompanhamento de agrupamentos (encontros) de indivíduos monitorados por dispositivos eletrônicos (tornozeleiras eletrônicas) produzindo dados com indicadores de riscos que auxiliarão nas tomadas de decisão por parte de órgãos de segurança pública ou agentes competentes. A fim de suportar um aumento considerável na quantidade de indivíduos monitorados, conforme indicativo resultante do montante exposto na Figura 1.2, a solução deve contemplar tecnologias que permitam a escalabilidade visando preservar o desempenho necessário. Desta forma, esta pesquisa propõe como objetivos específicos:

- a. Estabelecer algoritmos para detecção de agrupamentos de indivíduos monitorados através do processamento das coordenadas geográficas coletadas periodicamente e disponíveis na Central de Monitoramento;
- b. Estabelecer algoritmos para rastrear os agrupamentos registrando sua duração e quantidade de elementos;
- c. Estabelecer algoritmos para pesquisa e obtenção de dados complementares relativos a frequência dos encontros, horários mais comuns e regiões geográficas costumeiras dos encontros;
- d. Verificar a efetividade dos algoritmos com relação ao que foi proposto;
- e. Analisar o desempenho dos algoritmos através de cenários simulados em laboratório destacando as melhores alternativas para cumprimento do requisito sobre a janela de tempo de processamento;

## 1.3 Metodologia de Pesquisa

Do ponto de vista do objetivo geral, esta dissertação caracteriza-se, quanto a sua natureza como uma pesquisa aplicada ou tecnológica, uma vez que se destina à geração de conhecimentos para aplicação prática dirigidos à solução de problemas específicos.

Quanto aos procedimentos metodológicos, visando atingir os objetivos propostos, este trabalho de pesquisa foi dividido em seis fases, conforme apresentado a seguir:

**Fase 1:** Pesquisa bibliográfica sobre os conceitos pertinentes ao problema e sobre as técnicas acerca do monitoramento de dispositivos identificando também os principais trabalhos sobre detecção de agrupamentos;

**Fase 2:** Elencar eventuais restrições sobre a aplicação dos algoritmos abordados na solução do problema apresentado neste trabalho;

**Fase 3:** Abordar os principais desafios e definições na construção dos algoritmos de detecção de agrupamentos de pontos envolvendo coordenadas de dispositivos eletrônicos;

**Fase 4:** Desenvolver e implementar os algoritmos conforme os desafios apresentados;

**Fase 5:** Executar testes sobre os algoritmos utilizando uma massa de dados simulada com volume suficiente para evidenciar o funcionamento adequado e eventuais problemas de desempenho.

**Fase 6:** Analisar os resultados apresentados indicando, se for o caso, trabalhos futuros que possam expandir ou melhorar o assunto abordado aqui.

## 1.4 Contribuição do Trabalho

A contribuição focal do presente artigo é a concepção de um conjunto de algoritmos articulados, constituindo um modelo sistêmico capaz de processar os dados provenientes da rede para: 1) verificar proximidades entre pares de pontos (Detecção de Pares), 2) agrupar dispositivos em agrupamentos (Detecção de Grupos) e 3) computar dados complementares acerca do acompanhamento dos agrupamentos (Detecção de Riscos). Contribuições adicionais são descritas ao longo do trabalho quanto aos aspectos de implementação e de desempenho desses algoritmos.

Os algoritmos serão definidos de forma que possam, com poucas alterações, serem aplicáveis em outros contextos como, por exemplo, o monitoramento de espécie de animais em seu habitat natural. Neste caso, os dados acerca do agrupamento de animais podem indicar deslocamento de grupos, encontros e reencontros de grupos, acasalamentos com sua respectiva frequência, entre outros.

Dado a organização e modularidade dos algoritmos a serem desenvolvidos, os mesmos podem ter aproveitamento total ou parcial em outros contextos cuja a natureza envolva a detecção de grupos através de coordenadas geográficas ou o acompanhamento deste grupo através de seguidas amostras e coleções de pontos.

## 1.5 Organização do Trabalho

Este trabalho foi organizado em cinco capítulos, sendo este primeiro o de Introdução, com a finalidade de facilitar o entendimento desta pesquisa. Os demais capítulos estão organizados como descrito a seguir:

O Capítulo 2 aborda assuntos acerca dos conceitos utilizados além dos principais trabalhos disponíveis sobre detecção de agrupamentos e sistemas para monitoração e acompa-

nhamento de ações criminosas. Também realiza comparações sobre cada um dos algoritmos abordados ressaltando as principais restrições com relação ao problema proposto.

O Capítulo 3 apresenta a definição do problema e desafios relacionados aos requisitos relativos à solução. Também aborda o modelos sistêmico proposto e os algoritmos para solução do problema.

O Capítulo 4 apresenta os resultados de acordo com os cenários simulados, além da análise sobre sobre o processo realizado.

O Capítulo 5 conclui esta dissertação resumindo os resultados produzidos e sinalizando trabalhos futuros que podem dar sequência a esta pesquisa.

# Capítulo 2

## Fundamentação Teórica

Neste capítulo, alguns conceitos iniciais são apresentados como geoprocessamento, Sistemas de Informações Geográficas, bases de dados geográficos e processamento paralelo. Também destaca-se o estudo sobre Redes de Sensores Sem Fio e a Fusão de Dados (Data Fusion).

### 2.1 Geoprocessamento

Geoprocessamento é um conjunto de tecnologias de coleta, tratamento manipulação e apresentação de informações espaciais voltado para um objetivo específico [Francelino 2003].

Geoprocessamento ou geoinformática trata informações geográficas (ou espaciais) através de softwares e ferramentas específicas. Teve origem na década de 50 com o propósito de automatizar a produção de mapas utilizando técnicas matemáticas e computacionais. Devido às restrições tecnológicas da época, ainda era bastante incipiente e de custo elevado. Em 1989 o geoprocessamento foi reconhecido como uma "disciplina científica", logo após a fundação do Centro Nacional para Informação e Análise Geográfica (*National Centre for Geographical Information and Analysis – NCGIA*). Suas possíveis aplicações vão desde a cartografia, planejamento urbano, energia, comunicações, transportes até a análise de recursos naturais.

Na última década houve um massificação na utilização do geoprocessamento devido à ferramentas de fácil manipulação como por exemplo o *Google Earth* e o *Google Maps*. Com o uso massivo dos *smart phones* e a popularização de aplicativos de baixo custo que aproveitam os recursos de geolocalização dos dispositivos portáteis, a aplicação de geoprocessamento no cotidiano das pessoas tornou-se algo comum.

Sistemas de Informações Geográficas – SIG – surgiu na década de 70, a partir da evolução da tecnologia, inicialmente como ferramenta computacional para geoprocessamento. Realiza análises complexas ao integrar dados de diversas fontes e ao criar bancos de dados georre-

ferenciados tornando possível a produção automatizada de documentos cartográficos como mapas ([Francelino 2003]).

Através do estudo da Topologia, definiu-se estruturas geométricas como objetos que podem ser representados por propriedades e relações. Além da localização geográfica, tais objetos possuem propriedades como conectividade, contiguidade e pertinência, que fazem parte da natureza do objeto. Para que haja coerência nos objetos mostrados em um mapa, todas as suas coordenadas devem seguir um mesmo sistema de coordenadas.

A origem dos dados para compor o GIS pode variar podendo ser o resultado da digitalização de mapas, aplicação de fotos de satélites, coleta de coordenadas por dispositivos GPS, etc. Normalmente os dados coletados dessas diversas fontes são armazenados em um banco de dados geográfico.

A representação de dados espaciais seguem duas formas distintas:

### **2.1.0.1 Vetorial**

Na representação vetorial, os objetos que compõem o mapa são símbolos formados por pontos, linhas, polígonos e círculos. Os pontos são formados por uma par de números correspondentes à latitude e longitude indicando a posição geográfica dos mesmos. Linhas são formados por um conjunto de dois ou mais pontos interligados. Polígonos são formados por quatro ou mais pontos interligados onde o primeiro ponto tem exatamente a mesma coordenada do último ponto fechando graficamente o polígono. Diversos formatos diferentes podem ser representados no mapa através de um polígono como triângulos, quadrados, retângulos, trapézios, etc. Círculos são objetos compostos de uma coordenada que representa seu centro e um raio sobre o qual é projetado o círculo para representação no mapa ([Francelino 2003]).

Sobre objetos vetoriais dispersos em um mapa, o GIS é capaz de executar operações matemáticas para identificar objetos que se cruzam, que estão contidos em outros, que são contíguos, etc.

Por sua natureza, os objetos vetoriais são mais adequados para armazenamento em bancos de dados relacionais, desde que adaptados para tipos geográficos disponíveis na ferramenta escolhida.

### **2.1.0.2 Matricial (*Raster*)**

A representação matricial utiliza uma matriz de valores mapeados em linhas e colunas, onde cada célula contém um característica como cor ou profundidade. Esta matriz é traduzida como imagem onde cada célula representa um ponto (*pixel*) da imagem. A imagem é formada pelos pontos coloridos obtidos a partir da matriz.

A resolução da imagem/mapa tem especial importância na representação matricial.

Resoluções muito baixas podem reduzir a visibilidade de detalhes importantes impedindo a realização de análises mais precisas (Figura 2.1). Os diferentes tamanhos dos quadriculos nas imagens indicam a precisão que os objetos geográficos são apresentados. Na imagem da esquerda, por utilizar menor tamanho dos pontos apresenta uma melhor qualidade na visualização dos objetos. Em contrapartida, altas resoluções requerem maior espaço de armazenamento e maior esforço no processamento da imagem.

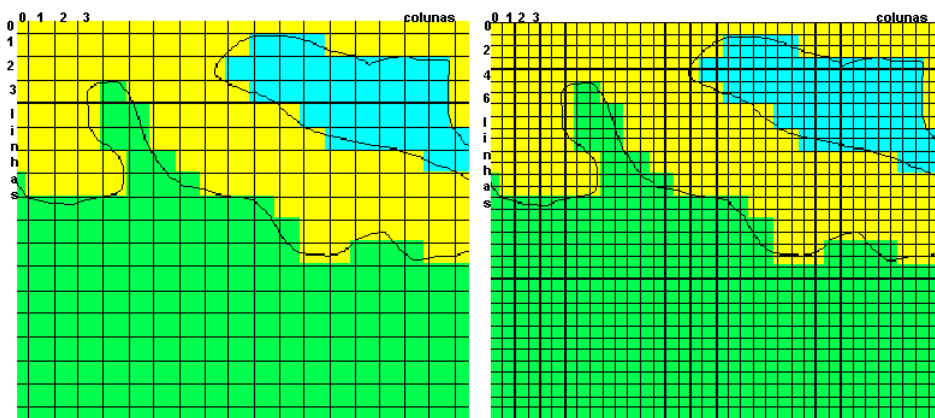


Figura 2.1: Diferenças entre resoluções matriciais. Fonte: [Câmara and Monteiro 2004]

Deve ser considerado também que para cada célula/pixel há uma escala de correspondência a ser calculada de forma a representar-se com relativa precisão (sem distorções) um mapa na tela ou impresso. Ao sobrepor um objeto vetorial sobre um raster, as proporções devem se manter compatíveis conforme a escala adotada mesmo com a aplicação de efeitos de *zoom*.

A Figura 2.2 ilustra as diferenças entre as representações vetoriais e matriciais. A representação vetorial, à esquerda da figura, é formada por pontos, linhas, polígonos, círculos, etc. A representação matricial, à direita da figura, é formada por pontos disposta numa matriz, onde o objeto é composto pela combinação desses pontos.

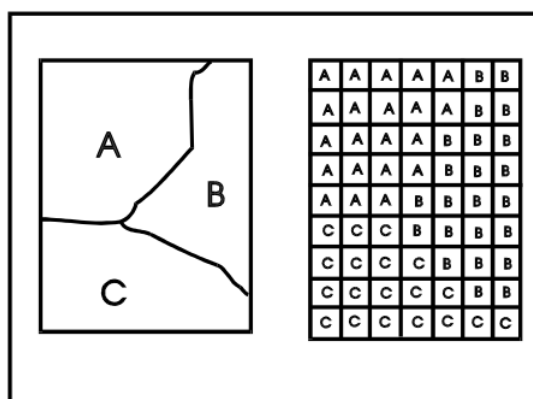


Figura 2.2: Comparação entre representação vetorial e matricial. Fonte: [Câmara and Monteiro 2004]

### **2.1.1 Sensoriamento Remoto**

Sensoriamento refere-se a um conjunto de tecnologias para aquisição de informação sobre um objeto geográfico através de sensores que não estão fisicamente em contato com o objeto. Um dos princípios básicos apoia-se na radiação eletromagnética (REM) do globo terrestre que é influenciada, entre outros fatores, pela energia solar. As características sobre como cada objeto observado absorve, reflete ou emite a radiação eletromagnética definem faixas espectrais que indicam suas propriedades. Esta tecnologia é especialmente importante para identificar os diversos tipos de região no planeta, assim como terra, mar, vegetação, etc. ([Francelino 2003]).

### **2.1.2 Sistema Global de Posicionamento (*Global Positioning System – GPS*)**

Os processos iniciais de levantamento de dados usavam técnicas cartográficas em campo através das medidas das distâncias e ângulos, onde através de cálculos permitiam gerar o mapa do terreno. Neste caso, além da baixa precisão, o tempo para realização do levantamento, entre medições e cálculos, era um fator a ser considerado no custo.

Com o surgimento dos primeiros satélites artificiais e o lançamento de dispositivos portáteis de GPS, geodestas perceberam rapidamente suas vantagens no levantamento topográfico. A aplicação do *GPS* mostrou-se mais prática e rápida para levantamento em campo.

Inicialmente criado para fins militares, tornou-se um dispositivo comum no meio civil, inclusive sendo embarcado em celulares, relógios e câmeras fotográficas.

A constelação de satélites dispõe de mais de 20 unidades o que proporciona uma cobertura de pelo menos quatro satélites em qualquer superfície da Terra. Esta cobertura contribui para a melhoria na precisão dos dispositivos.

### **2.1.3 Identificador de Referência Espacial (*Spatial Reference Identifier – SRID*)**

O globo terrestre não é dado por uma esfera perfeita. Antes de tudo, é necessário entender que a Terra tem um formato ligeiramente achatado. Além disso, existem partes no globo com altitudes extremas e outras regiões menos elevadas. Ao representar a crosta terrestre em forma de esfera em uma plataforma plana (mapa) as áreas e regiões serão distorcidas (projeção cartográfica). Para compensar esta distorção adotamos um sistema de coordenadas adequado a regiões específicas do mapa.

Em um determinado mapa, para que sua representação gráfica seja precisa, todos os objetos devem estar convertidos para o mesmo sistema de coordenadas geográficas de forma

que as camadas sobrepostas obedecem as devidas proporções. O sistema de coordenadas define como os dados geográficos são representados em um mapa plano e a unidade de medida utilizada (graus, pés, metros, etc).

Em fevereiro de 2005, com data limite de implantação finalizada em 2014, o Brasil adotou um novo referencial geodésico, chamado SIRGAS2000 (Sistema de Referência Geocêntrico para as Américas) compatível com WGS84 uma vez que ambos são da ordem de 1 centímetro [IBGE 2016]. O sistema WGS84 é o adotado pelo uso de sistema de posicionamento global GPS.

## 2.2 Bancos de dados geográficos

O avanço tecnológico possibilitou a integração de objetos geográficos e mapas. Os objetos geográficos, observadas as devidas adequações quanto ao tipo e estrutura, são passíveis de armazenamento através de sistemas gerenciadores de bancos de dados.

Normalmente, bancos de dados espaciais não contemplam a exibição gráfica dos elementos geográficos que armazena. Na maioria dos casos, ferramentas específicas para renderizar os mapas e seus objetos são fornecidos por outros fornecedores. Tais ferramentas possuem conectores apropriados para consultar o banco de dados espacial convertendo os dados obtidos em objetos gráficos visualizados em tela ou impressos.

Bancos de dados espaciais são sistemas que suportam tipos de dados específicos para representar objetos geométricos permitindo o armazenamento dos dados geométricos em forma de colunas em tabelas. Também provêm funções e índices especiais para manipulação e busca dos dados armazenados [Obe and Hsu 2015].

Os bancos híbridos, sistemas de bancos de dados relacionais (SGBDR) com adição de recursos para tratamento de dados geográficos, são amplamente aplicados. Os casos de maior notoriedade incluem o *PostGIS* para *PostgreSQL*, *Spatial* e *Locator* para *Oracle* e *CLR (.NET)* no *SQL Server*.

### 2.2.1 PostGIS sobre PostgreSQL

PostgreSQL é um popular sistema gerenciador de banco de dados (SGBD) objeto-relacional e de código aberto (*Open Source Database Management System – OSDBMS*) lançado em 1987. Tem evoluído a cada versão através da adição de melhorias contínuas e novas funcionalidades suportado por uma ativa comunidade de desenvolvedores. Até a publicação deste trabalho a 9.6 é a versão mais atual para baixar *download*.

O SGBD PostgreSQL oferece alguns recursos que são raros em outros bancos de dados como: i) suporte a várias linguagens de programação como Perl, Python, TCL, sendo a mais comum PL/PgSQL; ii) Tratamento nativo de matrizes (*arrays*); iii) Herança sobre tabelas; iv)



Forte aderência ao padrão ANSI-SQL (ANSI SQL 92-2003+ e parte do ANSI SQL 2006); v) Possibilidade de definição de novos tipos de dados definidos pelo usuário; vi) Suporte avançado a controle de transações; vii) Milhares de funções disponíveis nativamente ou por contribuição; e viii) Multiplataforma.

Sistemas gerenciadores de bancos de dados de código livre tem alcançado maturidade nos últimos anos podendo substituir produtos comerciais em muitas empresas, inclusive com elevada redução de custo [Feinberg and Adrian 2015].

PostGIS é uma extensão de código aberto para banco de dados PostgreSQL que adiciona novo conjunto de tipos de dados geográficos e funções específicas para manipulação de dados geográficos. Adiciona um conjunto de funções à linguagem estruturada de consultas (Structured Query Language – SQL) para trabalhar com objetos geométricos em uma base de dados. Desta forma, funções espaciais permitem criar e modificar objetos no espaço. Esta análise espacial é frequentemente referenciada como processamento geométrico ou espacial [Obe and Hsu 2015].

O Consórcio Geoespacial Aberto (*Open Geospatial Consortium – OGC* – mantém o padrão OGC em diversas especificações para dados espaciais e geográficos e consultas (*queries*) de dados geoespaciais. A Fundação Geoespacial de Código Aberto (*Open Source Geospatial Foundation – OSGeo*) é uma organização sem fins lucrativos criada para desenvolver o PostGIS de forma colaborativa e promover sua utilização. Os dois têm a missão em comum de desenvolver ferramentas GIS e fazê-las disponíveis para o público em geral através de especificação de padrões abertos [OSGeo 2016].

Um dos padrões OGC estabelecidos é a representação textual dos objetos geoespaciais denominado texto bem conhecido (*Well-known text – WKT*). Por exemplo, a representação de um ponto geográfico pode ser dada pela texto `POINT(-77.036548 38.895108)`. Embora seja geralmente utilizado o padrão *WKT* para visualizar objetos, os mesmos são armazenados no banco de dados de forma binária. Então é comum inserir os objetos no banco de dados utilizando funções para converter *WKT* para binário. Da mesma forma, ao recuperar o objeto, existem funções que convertem o binário recuperado em texto legível.

A opção pelo PostGIS/PostgreSQL neste trabalho tem, entre outros fatores, cunho financeiro, uma vez que apresentam custos reduzidos, como também cunho técnico, dado a flexibilidade oferecida por essa solução (Criação de novos tipos de dados e funções definidas pelo usuário). Também levou-se em consideração o amplo suporte disponível na Internet tanto para quem está iniciando na área de GIS como para quem possui conhecimentos avançados.

### **2.2.1.1 Atributos georeferenciados**

Atributos georeferenciados são objetos geoespaciais que o banco de dados pode armazenar e manipular. O atributo mais simples armazenado corresponde a um ponto que

refere-se a uma coordenada geográfica em um mapa ou um ponto específico em uma área geométrica. O ponto é composto de um par de números indicando respectivamente longitude e latitude. Um exemplo de especificação de ponto em WKT seria `POINT (-77.036548 38.895108)`. O ponto, assim como outros atributos georeferenciados, pode ter um atributo adicional referente ao sistema referencial SRID.

Outro atributo georeferenciado suportado pelo PostGIS é a linha (`LINESTRINGS`) que é composta por uma sequência de pontos interligados. Uma representação de linha em WKT seria `LINESTRING(-14 21, 0 0, 35 26)`. Em um mapa, a linha poderia representar o curso de um rio ou uma rodovia, por exemplo.

O polígono é uma linha formada pontos onde o primeiro da sequência e o último possuem exatamente a mesma coordenada. A linha também pode formar a fronteira externa de um polígono (*exterior ring*) como por exemplo `POLYGON((0 1, 1 -1, -1 -1, 0 1))`. Também é possível incluir buracos nos polígonos (*inner rings*) conferindo maior flexibilidade na delimitação de áreas mais complexas.

É possível armazenar atributos georeferenciados combinando pontos, linhas e polígonos (*multipoints, multilinestrings, multipolygons* e *geometrycollections*) permitindo o armazenamento de objetos geométricos complexos.

PostGIS oferece mais de 300 funções e operadores para manipular atributos georeferenciados. Entre as operações executadas pelas funções, podemos citar alguns grupos: i) cálculo de distância entre dois objetos (objetos mais próximos ou mais longe); ii) cálculo da área de polígonos; iii) objetos que estão contidos ou se cruzam (*overlap*) com outros objetos (interseções); iv) cálculo do perímetro do objeto; v) conversão entre os diversos formatos de especificação de objetos geoespaciais; vi) validação de objetos;

Os tipos de dados geográficos (*geography*), diferente dos tipos de dados geométricos (*geometry*), considera o formato da terra como uma esferoide em um sistema SRID 4326. Tipos geográficos esperam dados especificados em graus no padrão WGS84 e retornam o resultado em metros. Se os dados estão em diferente sistema referencial, é necessário convertê-lo para SRID comum. Os cálculos de distância e áreas para atributos geográficos usando esferoide terrestre no padrão WGS84 também suportam o modelo esférico, mais rápido porém menos preciso, com o raio estipulado em 6370986 metros [Obe and Hsu 2015].

## 2.3 Computação Paralela

O estudo da computação paralela é relativamente antigo, porém tem recebido mais atenção nos últimos anos principalmente em função da limitação física que impede o aumento da frequência dos processadores impelindo os fabricantes a adotarem a solução de múltiplos núcleos. Mesmo máquinas de único processador utilizadas em conjunto com outras, têm a capacidade de formar agrupamentos (*cluster*) para processamento paralelo.

A computação paralela se aplica a problemas que podem ser divididos e executados concorrentemente através da distribuição das tarefas em filas reduzindo o tempo de execução [Quinn 1986]. Cada fila pode ser representada por nó de processamento, processador ou núcleo de processador (*threads*).

Na computação paralela, partes de programas sequenciais, onde as instruções são executadas uma de cada vez, são substituídas por blocos de programas paralelos executados de forma concorrente, para resolver o mesmo problema inicial e produzir o mesmo resultado. Geralmente apresenta um custo maior (aumento da complexidade de infraestrutura ou no maior esforço de codificação do algoritmo), porém com desempenho melhor. Se o contexto não requer o ganho de desempenho entregue pelo paralelismo, não há motivos para aumentar o custo aplicando tal arquitetura [OpenShaw and Turton 2014].

A arquitetura paralela pode utilizar um único computador com múltiplos processadores, diversos computadores interligados em rede, hardware especializado ou qualquer combinação destes. A programação paralela traz alguns potenciais problemas não existentes na programação sequencial como a necessidade de sincronização ou comunicação entre os processos.

Para que um programa possa ser paralelizado, é necessário identificar tarefas que possam ser executadas concorrentemente em mais de um processador ou nó. O programa é favorável à paralelização quando seu algoritmo permiti a divisão do processamento em blocos mantendo o mesmo resultado final. Um algoritmo é escalável se o nível de paralelismo pode aumentar no mínimo linearmente com o tamanho do problema. Ou seja, uma arquitetura é dita escalável se continua a alcançar o mesmo desempenho por processador, mesmo com o aumento do problema, apenas adicionando novos nós ao sistema.

O aumento da velocidade do processamento com o paralelismo deveria ser linear de forma que dobrando a quantidade de processadores, como resultado esperado, reduziria o tempo de execução pela metade. Entretanto, poucos algoritmos possuem 100% de seu código paralelizável, ou seja, deve-se considerar a parcela do algoritmo que não é paralelizável. A equação 2.1 representa o tempo total  $T_N$  em função do número de processadores  $N$ , onde  $T_S$  representa o tempo gasto na parte não paralelizável e  $T_P$  o tempo total gasto na parte paralelizável do algoritmo. O aumento de velocidade é quase linear para poucos elementos de processamento, tendendo a um valor constante para uma grande quantidade de elementos. Baseado nesta ideia, [Amdahl 1967] afirma que a escalabilidade em um sistema paralelo possui um limite máximo e isto pode colocar em questão o futuro do processamento paralelo [OpenShaw and Turton 2014].

$$T_N = T_S + \left(\frac{T_P}{N}\right) \quad (2.1)$$

Embora haja um limite na eficiência do paralelismo em função da adição de mais processadores, na prática temos visto que esta técnica tem sido ainda largamente utilizada e com amplo espaço para crescimento. Um bom exemplo disso é o lançamento contínuo de

supercomputadores cada vez mais eficientes baseados nesta tecnologia.

O segundo indicador a ser considerado é a eficiência (*efficiency*) onde a atual aceleração é comparada com à aceleração ideal. Desta forma, teríamos uma eficiência considerada ideal (máxima) quando o valor de *Speedup* fosse igual à quantidade de processadores  $N$ . Encontraríamos tal cenário em um ambiente com algoritmo totalmente paralelizado e com nenhuma perda de paralelismo em outros fatores como a comunicação entre processos e o tempo de acesso à memória. Dada a dificuldade de se obter tal cenário, a eficiência pode se aproximar, mas dificilmente alcançará o valor ideal.

Uma vez que a aceleração ideal é igual ao número de processadores  $N$ , então a eficiência  $\varepsilon$  pode ser representada pela equação 2.2, onde  $N$  é a quantidade de processadores. Uma vez que  $(T_N \times N)$  será sempre menor ou igual a  $T_1$  então  $0 \leq \varepsilon \leq 1$  e quanto mais próximo o resultado de 1, maior a eficiência do sistema [OpenShaw and Turton 2014].

$$\varepsilon = \frac{T_1}{(T_N \times N)} \quad (2.2)$$

Conforme as Equações 2.1 e 2.2, se a parte serial do algoritmo for bem maior que a parte paralelizável, então a eficiência tende a ser muito baixa. Partindo desta linha, o desenvolvimento do algoritmo deve buscar o máximo de blocos paralelizáveis afim de obter uma maior eficiência da paralelização.

### 2.3.1 Classificação da computação paralela

Segundo a taxonomia de [Flynn 1972], um sistema pode ser classificado em 4 categorias diferentes de acordo com as instruções (simples ou múltiplas) e com os dados (simples ou múltiplos). Atualmente, podemos encontrar sistemas em mais de uma categoria, porém a classificação de Flynn auxilia o entendimento das arquiteturas utilizadas.

#### 2.3.1.1 Instruções Simples e Dados Simples (*Single Instruction, Single Data – SISD*)

Sistemas *SISD* são ambientes cujos programas não empregam o paralelismo. Nesta categoria estão todos os programas onde as instruções são executadas sequencialmente, uma após a outra onerando um único núcleo processador.

#### 2.3.1.2 Instruções Simples e Múltiplos Dados (*Single Instruction, Multiple Data – SIMD*)

Mesmo conjunto de instruções executado por diversos processadores sobre diferentes partes do mesmo conjunto de dados (Figura 2.3). O resultado total é apresentado quando todos os processadores concluírem sua parte da tarefa.

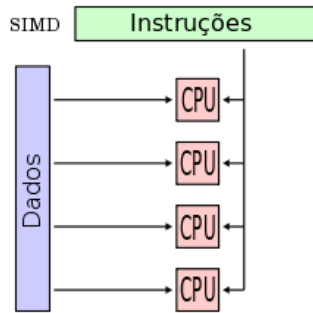


Figura 2.3: Single Instruction, Multiple Data – SIMD.

### 2.3.1.3 Múltiplas Instruções e Dados Simples (*Multiple Instruction, Single Data – MISD*)

*MISD* são casos raros de instruções diferentes sobre os mesmos dados. Pode ser aplicado em situações em que uma grande tabela de dados deve ser lida por vários processos paralelos sendo que cada um desses processos tenha algoritmo diferente produzindo diferentes resultados.

### 2.3.1.4 Múltiplas Instruções e Múltiplos Dados (*Multiple Instruction, Multiple Data – MIMD*)

Método mais comum adotado em arquitetura de processamento paralelo, *MIMD* não é obrigado a trabalhar de forma altamente sincronizada, em vez disso o sincronismo deve ocorrer apenas durante a execução do programa [OpenShaw and Turton 2014]. A Figura 2.4 ilustra o funcionamento deste tipo de hardware.

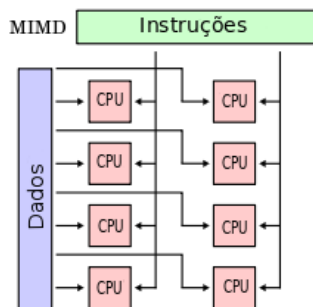


Figura 2.4: Multiple Instruction, Multiple Data – MIMD.

A Figura 2.5 mostra um caso de *MIMD* onde há a execução paralela de tarefas de tamanhos diferentes combinando múltiplas precedências.

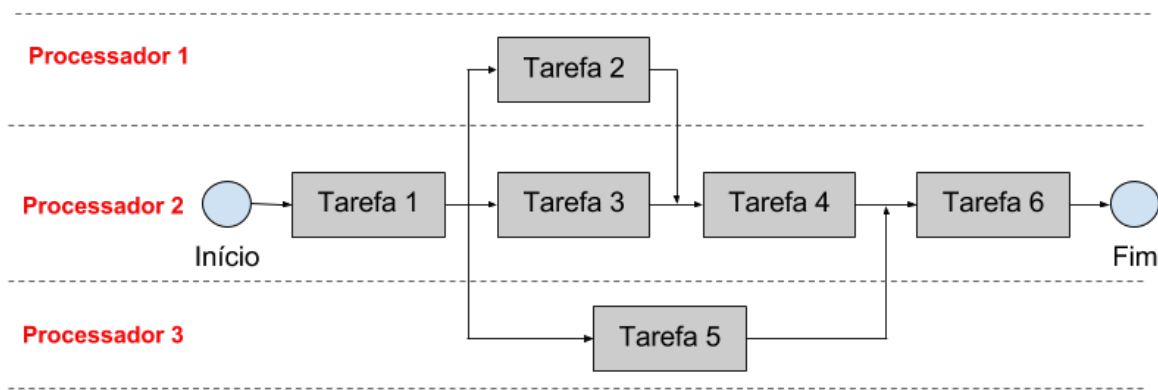


Figura 2.5: Algoritmo executado em paralelo com 3 processadores e MIMD.

### 2.3.2 Escalonamento

O escalonamento corresponde ao processo de distribuir as tarefas para os vários processadores disponíveis atuando como um orquestrador. A política de distribuição pode ser: i) estática – onde o processador a executar a tarefa é definido antes de iniciar o processamento; ou ii) dinâmica – onde o processador para execução da tarefa é definido durante a execução do processo como todo.

O escalonamento é particularmente importante no sentido de aumentar a eficiência do sistema paralelo. Optar por processadores mais ociosos no momento de delegar a execução da atividade é uma forma de aproveitar melhor os recursos disponíveis reduzindo o tempo de execução. Sistemas operacionais modernos em computadores com múltiplos processadores utilizam esta técnica para distribuir os processos.

O desempenho é alcançado no paralelismo quando um número de comandos são executados simultaneamente, aproveitando melhor o processador ou nó. Neste caso é preciso considerar o grau de dependência entre os processos.

Em cenários com SIMD, como o volume de dados para cada tarefa é constante e o número de processadores é previamente conhecido, a política de distribuição de tarefas pode ser estática. Cenários com processadores de diferentes capacidades ou tarefas com tamanhos diferentes tornam a distribuição das tarefas mais complexa exigindo certa inteligência no processo escalonador.

Quando a quantidade de tarefas prontas para execução é menor ou igual a quantidade de nós temos uma execução paralela real, onde cada nó dedica-se totalmente à tarefa que lhe foi delegada. Quando há mais tarefas que processadores, é necessário: i) aguardar a liberação de algum processador para delegar-lhe nova tarefa; ou ii) delegar a tarefa ao processador para que o mesmo administre a execução concomitante das diversas atividades.

No segundo caso, processadores modernos trazem recursos para atender requisições concomitantes da forma mais homogênea possível utilizando o processo de troca de contexto. Com isto, atendem virtualmente mais tarefas intercalando seu tempo com as demandas em

atendimento.

### 2.3.3 Memória compartilhada e memória distribuída

A memória compartilhada é uma estrutura contida em um computador onde todos os processadores acessam os dados através de um único canal (*bus*). Todos os endereços da memória global estão disponíveis da mesma forma para todos os processadores. Na prática, os blocos de dados podem até estar separados em porções distintas da memória global, mas os processadores entram numa fila para obter ou gravar algum dado na memória (*bus*). Esta restrição é compensado pelos *caches* internos dos processadores. Apesar desta restrição, a programação para este tipo de plataforma é mais simples por não haver a necessidade de envio de mensagens de comunicação entre processos, como no caso de memórias distribuídas.

No caso da memória distribuída, as porções de memória encontra-se em computadores distintos. Reduz-se assim a concorrência de acesso a memória pelos processadores. Porém, em caso de necessidade de acesso a dados em outro computador, uma mensagem precisa ser elaborada e trafegada pela rede com a solicitação e a respectiva resposta, podendo representar um gargalo neste caso. Isto pode ser agravado, a depender da frequência e do tamanho das mensagens e também da velocidade da rede.

### 2.3.4 Multiprocessadores Simétricos (*Symmetric MultiProcessors – SMP*)

A restrição sobre o aumento da velocidade do processador em termos de frequência (*clock*) fez com que os fabricantes de processadores investissem em *chips* com múltiplos núcleos. Esta tecnologia foi aplicada à servidores de baixa plataforma. Porém, é atualmente comum encontrar muitas estações de trabalho (*desktops*), *notebooks* e *smartphones* com processadores de vários núcleos (*cores*) permitindo a execução de algoritmos paralelos com menor tempo total de execução.

Estes equipamentos são categorizados como Multiprocessadores Simétricos, pois compartilham algumas características comuns: i) Memória principal centralizada e compartilhada; ii) Todos os processadores simétricos, ou seja, de igual capacidade; iii) Os processadores compartilham o canal de acesso aos recursos; iv) Geralmente o sistema operacional é responsável por distribuir e administrar os processos sobre os processadores.

Devido à limitação de acesso à memória compartilhada por um único canal, a adição de muitos processadores em um mesmo sistema *SMP* pode gerar retenções. Para reduzir este impacto, os fabricantes de processadores aumentaram a memória interna dedicada a cada processador (*cache*).

## 2.4 Computação Distribuída

Computação distribuída é um tipo de processamento paralelo em que as tarefas são divididas entre um conjunto de computadores homogêneos ou heterogêneos, podendo estar dispostos em um mesmo ambiente físico ou geograficamente dispersos, desde que interligados em uma mesma rede (conectividade). Portanto, computação distribuída é mais abrangente e universal do que a computação paralela.

A topologia da rede de interconexão é um requisito crítico para o desempenho do sistema, influenciado pela velocidade, latência de comunicação e sobrecargas de envio e recebimento de mensagens. A computação distribuída tende a ser uma melhor opção em volumes mais elevados de dados por conta da quantidade de processadores (ou nós) que pode ser adicionada ao sistema.

### 2.4.1 Processadores Massivamente Paralelos (*Massively Parallel Processors – MPP*)

Geralmente a computação distribuída faz uso de sistemas de Processadores Massivamente Paralelos, que são dotados de grande quantidade de processadores agregados. Ao contrário de SMP, ambientes com esta arquitetura possuem características como: i) Memória distribuída nos nós; ii) Troca de mensagens por rede de interconexão; iii) Processadores e nós heterogêneos; iv) Cada nó executa seu próprio sistema operacional; e v) Recursos locais não são necessariamente compartilhados entre os nós [OpenShaw and Turton 2014].

Diferentes topologias podem ser empregadas, inclusive mesclando nós de um ambiente *MPP* com configuração *SMP* tirando proveito dos dois ambientes.

Vale salientar que embora *MPP* ofereça maior poder computacional e maior flexibilidade, também traz maior complexidade. Esta complexidade exige um melhor planejamento na disposição da infraestrutura bem como torna o desenvolvimento dos programas mais oneroso em termos de esforço de codificação.

### 2.4.2 Grupo de computadores (*Clusters*)

Conjunto de computadores interligados por uma rede local com características básicas: i) geograficamente centralizados; ii) ausência de um escalonador centralizado; e iii) baixo custo. Geralmente os nós do *cluster* trabalham sincronizados e com objetivo comum.

### 2.4.3 Computação em Grade (*Grid Computing*)

A grande característica da computação em grade é a disposição dos seus nós podendo estar conectados localmente ou geograficamente distantes formando uma única máquina vir-



tual. Os nós normalmente apresentam diferenças tanto de hardware como de software incluindo o sistema operacional (elevada heterogeneidade), pois esta arquitetura se baseia no *MPP*. O controle das tarefas é totalmente distribuído contando com um escalonador para coordenar a execução das atividades.

Também apresenta forte tolerância a falha. Alguns nós podem não estar disponíveis o tempo todo requerendo que as atividades interrompidas sejam concluídas em outros nós. Com a quantidade de nós costuma ser grande, há sempre disponibilidade parcial do sistema.

#### 2.4.4 Geoprocessamento Paralelo

A pesquisa sobre geoprocessamento paralelo e outras formas de obter melhor desempenho no processamento geográfico é relativamente antiga. Em 1975, [Peucker and Douglas 1975] propôs três algoritmos para extrair características de terrenos, onde cada processo era independente dos outros implementando operações sobre dados vizinhos. O método de [Dangermond and Morehouse 1987] é diferente por discutir o modelo de resolução de problema pelo recurso de condutores (*pipeline*) e como tirar vantagem de subdivisão espacial por hierarquia. Outras pesquisas focaram a utilização de supercomputadores vetoriais, que são projetados para processamento de grande volume de dados em alta velocidade destinado à computação científica. Estes supercomputadores são dotados de processadores com instruções específicas que operam sobre matrizes unidimensionais de dados chamados vetores [Ding and Densham 1996].

Um dos principais desafios na área de geoprocessamento paralelo é a decomposição dos algoritmos utilizados, também chamado de particionamento, permitindo distribuição do processamento, principalmente nos casos de SIMD. Cada algoritmo possui suas próprias características e problemas a serem resolvidos para permitir a paralelização.

Uma alternativa seria considerar uma região geográfica de interesse e dividi-la em sub-regiões (ou subdomínio) dispostos na forma um mosaico (*tessellation*). Estas subáreas não têm necessariamente um tamanho regular ou formato fixo. Cada uma destas sub-regiões possuem uma quantidade de zero ou mais pontos.

A divisão em quadrados e retângulos também é comum no tratamento de GIS e modelagem espacial. Um dos métodos mais populares é o *quadtree*, que utiliza particionamentos recursivos do espaço em conjuntos de quadrados (ou retângulos) semelhante ao que é aplicado em árvores de busca binária [Samet 1990].

Para bancos de dados espaciais, devido as coordenadas dispostas em latitudes e longitudes, o formato quadrado ou retangular representa uma forma eficiente de aproveitamento do processamento.

É interessante analisar a natureza dos objetos geográficos dispostos em uma área geográfica antes do processamento. Para tanto, a Tabela 2.1 ilustra os quatro tipos propostos por [Ding et al. 1992].

Características	Regular	Irregular
Homogêneo	Tipo 1	Tipo 2
Heterogêneo	Tipo 3	Tipo 4

Tabela 2.1: Classificação do problema espacial conforme a característica. Adaptado de [Ding et al. 1992]

A característica relativa à diversidade de tipos de objetos indica se é homogêneo ou heterogêneo. Enquanto a dispersão ou concentração dos objetos no espaço indicam respectivamente a regularidade ou irregularidade.

O Tipo 1, por ser homogêneo e regular, representa um problema onde a divisão do espaço geográfico pode ser dividida de forma igualitária e o processamento realizado de forma mais simples. O Tipo 2 possui membros de mesmo tipo porém dispersos de forma irregular no espaço dificultando o particionamento do espaço. Tipo 3 possui tipos diferentes de elementos mas com uma dispersão uniforme dos mesmos permitindo um particionamento mais fácil, mas prejudicando o balanceamento de carga, considerando que os tipos diferentes requererão níveis de processamento diferente. Por último, o Tipo 4 apresenta o pior caso pois apresenta tanto a dificuldade no balanceamento de carga quanto à divisão dos espaços, trazendo em geral, maior ineficiência no processamento.

Para complementar o processamento proposto, geralmente o algoritmo referente ao geoprocessamento precisa tratar parte dos objetos das áreas vizinhas. Para isso, a comunicação entre processos é requerida para intercâmbio de dados, e isto pode degradar o desempenho do processamento paralelo com todo. Tal problema é agravado no caso de *MIMD* em cenário com memória distribuída [Ding and Densham 1996]. Uma alternativa à solução de comunicação entre processos é a sobreposição de subáreas que pode variar conforme os tipos descritos a seguir ([Tomlin 1990]).

#### 2.4.4.1 Modelagem espacial com escopo local

O resultado final consiste da junção simples dos resultados locais, individuais de cada sub-região. Neste caso, dispensa-se a necessidade de sobrepor uma sub-região sobre a outra. O processamento final costuma ser produzido pela união simples dos resultados parciais.

#### 2.4.4.2 Modelagem espacial com escopo de vizinhança

O algoritmo deverá considerar a porção de área que lhe foi atribuída, mais as demais áreas subjacentes. Esta modelagem poderá implicar em redundância de processamento. Além disso, o resultado final pode precisar de algum tratamento para descartar eventuais duplicação.

#### **2.4.4.3 Modelagem espacial com escopo regional**

Neste caso, a sub-região sobrepõe uma parcela das sub-regiões vizinhas. Assim como o caso anterior, o processamento exige um processamento adicional ao final para tratar o resultado. A quantidade de área sobreposta às sub-regiões vizinhas depende do escopo das operações realizadas e do nível de dependência dos dados.

#### **2.4.4.4 Modelagem espacial com escopo global**

Todos os membros (sub-regiões) são usadas para o computar o resultado final. Este cenário é que apresenta a maior dependência horizontal de dados, aumentando também a dependência na troca de mensagens.

### **2.5 Redes de Sensores Distribuídos sem Fio (*Wireless Sensor Network - WSN*)**

Avanços recentes na tecnologia de sistemas de micro e eletromecânicos, comunicação sem fio, e eletrônica digital têm permitido o desenvolvimento de nós de sensores multifuncionais de baixo custo e consumo. Redes de sensores fornecem ao usuário uma inteligência e uma melhor compreensão acerca do ambiente monitorado. Segundo [Akyildiz et al. 2002], a expectativa é que no futuro, redes de sensores sem fio farão parte integral de nossa vida, mais que a atual presença de computadores pessoais.

Sensores podem monitorar uma série de diferentes fenômenos como temperatura, umidade, deslocamentos, posição, luminosidade, pressão, ruídos, presença de objetos, velocidade, etc. As aplicações de redes de sensores sem fio abrangem diversas áreas como saúde, militar, segurança, monitoramento de ambientes, automação residencial, etc. Por exemplo, dados biológicos acerca de um paciente podem ser remotamente monitorados por um médico. Em outro exemplo, sensores podem medir ou detectar a concentração de agentes químicos na água ou no ar para identificar áreas mais poluídas e evolução da poluição no meio ambiente (análise temporal).

Redes de sensores são geralmente compostas por um grande número de nós densamente distribuídos dentro ou bem próximos do fenômeno observado. Neste contexto, os nós não necessitam de um planejamento prévio quanto à fixação dos dispositivos em uma posição, tornando a implementação mais ágil. Em contrapartida, os sensores devem fazer uso de protocolos de rede e algoritmos com capacidade de se auto-organizar.

Dada a crescente capacidade de processamento embarcado nos dispositivos sensores, em vez de enviar dados não tratados para o nó responsável pela fusão dos dados, os sensores tem a capacidade de executar computações simples e transmitir somente dados requeridos e dados parcialmente processados otimizando a utilização da rede.

## 2.5.1 Fatores que influenciam o projeto de redes de sensores

Alguns fatores devem ser levados em consideração em seu projeto, entre eles a tolerância a falha, escalabilidade, custo de produção, ambiente operacional, topologia de rede, restrições de *hardware*, meio de transmissão e consumo de energia. Existem muitas pesquisas sobre estes temas, porém nenhum que integre a visão de todos esses fatores orientados ao projeto de rede de sensores e nós sensores ([Akyildiz et al. 2002]).

- a. Tolerância a falhas - Alguns sensores podem parar por falha diversas (*hardware*, esgotamento da bateria, bloqueio de sinal, etc.). Tais falhas porém não deveriam afetar a rede de sensores com um todo. Tolerância a falha é a habilidade de se manter as funcionalidades da rede de sensores sem interrupção devido a falhas em nós específicos. Para tanto, algoritmos e protocolos devem ser projetados para suportar tais falhas, de acordo com a criticidade da aplicação.
- b. Escalabilidade - O número de sensores podem alcançar a ordem de centenas, milhares ou milhões, dependendo do tipo de aplicação. O projeto da rede de sensores deve estar preparado para lidar com a quantidade de sensores na rede. Também deve-se considerar a alta densidade de nós em uma região e a proximidade entre os mesmos. Este fator tem implicação no alcance a rede e no número de canais de comunicação disponíveis.
- c. Custos de produção - Mesmo considerando a redução dos preços de *hardware*, a grande quantidade de nós sensores a serem adquiridos pode inviabilizar o projeto. Deve-se neste caso, aproveitar a aquisição de um grande lote e negociar valores suficientemente baixos para permitir a execução do projeto dentro de um orçamento aceitável.
- d. Restrições de *hardware* - Basicamente um nó sensor é composto de um sensor propriamente dito, uma unidade de processamento, uma unidade de comunicação e uma bateria (*power unit*). Eventualmente, deverá ter também uma unidade de geolocalização, uma fonte de energia alternativa (placa de energia solar, por exemplo) e um mobilizador (mecanismo para deslocar o sensor).
- e. Tamanho - Todos estes componentes devem ter tamanho tal que caibam em uma caixa (*case*) de tamanho adequado ao ambiente a ser monitorado. Cada um dos componentes do nó sensor possuem suas próprias restrições que ainda podem variar em função de suas especificações.
- f. Topologia de rede - Tratar nós inacessíveis ou desvinculados da rede faz da manutenção da topologia uma tarefa desafiadora. O protocolo deve estar preparado para tratar milhares ou milhões de nós sensores na mesma rede. Além disso, a proximidade entre os nós impõe restrições com relação à disputa por canais disponíveis.

- g. Ambiente operacional - Redes de sensores, dependendo de sua aplicação, devem estar preparados para funcionar de forma autônoma, contornando localmente eventuais falhas ou acumulando um volume de dados para posterior transmissão, por exemplo.
- h. Meio de transmissão - Meios de comunicação numa rede de sensores podem ser por rádio e infravermelho. Em caso de comunicação por rádio, as bandas de frequências disponíveis pela legislação local devem ser observadas. A escolha do meio de transmissão deve levar em consideração, entre outros aspectos, o consumo de energia. Infravermelho é uma alternativa de fácil implementação e que não sofre interferência elétrica, porém exige um campo de visada entre o transmissor e o receptor.
- i. Consumo de energia - No caso de bateria, seu tamanho e tecnologia devem ser escolhidos com bastante cautela de forma a manter o dispositivo ligado por mais tempo. Outros meios de geração de energia podem ser usados como por células fotovoltaicas. Cada componente adicional no circuito representa uma parcela a mais no consumo de energia, exigindo uma boa análise de desempenho no projeto da rede de sensores.

## 2.6 Fusão de Dados (*Data Fusion*)

Fusão de dados pode ser definido como uma combinação de múltiplas fontes para obter informações aperfeiçoadas ou melhoradas para prover informações com menor custo, com maior qualidade e com informações relevantes [Castanedo 2013]. Fusão de dados tem sido fortemente empregada em ambientes de múltiplos sensores para fundir e agregar dados de diferentes fontes, aplicável também em outras áreas.

### 2.6.1 Classificação de Fusão de Dados

As técnicas de Fusão de Dados podem ser classificadas em: i) Associação de dados, ii) Estimativa de estado, e iii) Fusão de decisão. Por ser uma área multidisciplinar torna-se difícil o estabelecimento preciso das possíveis classificações aplicadas a Fusão de Dados. Desta forma, quanto ao relacionamento entre as fontes de dados, Fusão de Dados pode ainda ser classificado em: i) complementar, ii) redundante, e iii) cooperativo. Neste caso, este trabalho está mais voltado para "complementar", uma vez que esta categoria é definida pela situação em que as informações de múltiplas fontes fornecem diferentes parte de dados que compõem a cena para obter uma informação global completa.

Com relação ao nível de abstração, podemos ainda classificar em: i) nível de sinal, ii) nível de pixel, iii) característico, e iv) símbolo [Luo et al. 2002], sendo a primeira opção a que mais define o processamento abordado neste trabalho.

Uma das classificações mais difundidas é a proposta pela *Joint Directors of Laboratories – JDL* – em parceria com do *Department of Defense – DoD* – que defende 5 níveis

de processamento dividido em 2 grupos: i) fusão de baixo nível e ii) fusão de alto nível, cada um destes compostos de i) fontes, ii) interação homem / computador, e iii) sistema de gerenciamento de banco de dados.

Estes 2 grupos são ainda classificados em 5 categorias, cuja representação está melhor detalhada na figura 2.6.

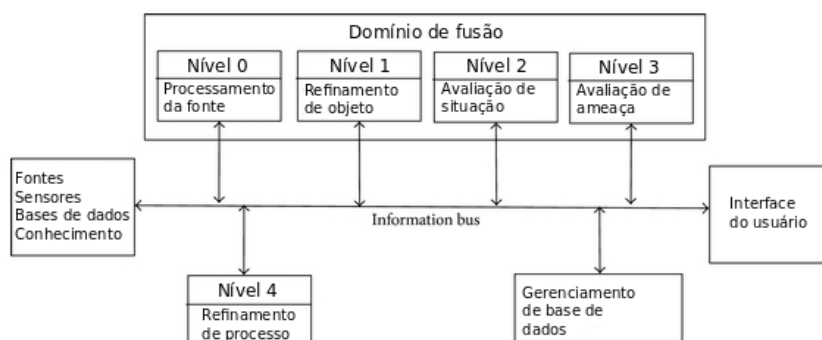


Figura 2.6: Framework de Fusão de Dados JDL. Fonte: [Steinberg et al. 1999]

Existem várias outras classificações disponíveis em uma grande quantidade de publicações, porém separamos aqui as mais agregam novo valor a este trabalho, o que não as tornam menos importante em outros contextos.

## 2.6.2 Técnicas de Associação de Dados

A Associação de Dados (*Data Association*) tem por objetivo estabelecer um conjunto de observações ou medições geradas pelo mesmo objeto através do tempo gerando uma trilha ou trajetória. O problema de associação de dados compreende os passos: i) observações do sensor (dado coletado) são recebidas pelo nó de fusão em período de tempo regular, ii) uma observação pode não ser fornecida em um período específico (falha) e precisa ser tratada, iii) algumas observações podem ser reais e outros resultados de alguma falha (ruídos), e iv) para quaisquer objetos observados no decorrer do tempo, não se sabe que direção será tomada [Castanedo 2013].

O problema em associar cada medida (observação) válida com o objeto-alvo apropriado ou então descartar uma determinada medida com sendo uma “perturbação” (ruído) é conhecido como Associação de Dados, Associação de Medição (*Measurement Association*) ou ainda Correlação de Dados (*Data Correlation*) [Premebida 2005].

A Figura 2.7 resume o processamento de associação de dados. Supondo um situação com múltiplos quadros (*frames*) associados e assume-se que há  $M$  pontos detectados em todos os  $n$  quadros, então o número possível de conjuntos seria  $(M!)^{n-1}$ , onde apenas um desses conjuntos representa o movimento correto de  $M$  pontos [Castanedo 2013].

A associação de dados assume que os dados que chegam até seu processamento foram

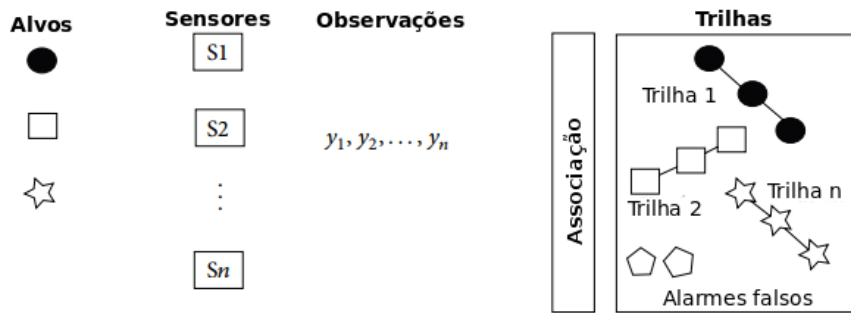


Figura 2.7: Modelo conceitual do processamento de associação de dados. Fonte: [Castanedo 2013]

devidamente tratados através de filtragem, segmentação, extração de pontos/características, etc.

### 2.6.3 Considerações acerca da Fusão de Dados e o trabalho proposto

Uma das áreas amplamente aplicadas é o processamento de imagens voltadas à robótica. A tarefa de seguimento (*tracking*) de múltiplos alvos na mesma vizinhança (dentro da elipse de erro), e ainda os “falsos” objetos e perturbações (ruídos), podem tornar o ambiente significativamente complicado. A associação de medidas quando há vários objetos-alvo precisa ser realizada considerando-se todos os objetos simultaneamente, isso porque caso uma medida tenha sido considerada como tendo origem de mais de um objeto, o cálculo da probabilidade de associação torna-se altamente acoplado (interdependente) e esses objetos não podem ser então considerados separadamente [Premebida 2005].

Fusão de dados é um conceito que envolve diversos métodos e não apenas uma técnica. É utilizada em diversas áreas do conhecimento e encontra-se em estágio de evolução, avançando na medida que novos sensores são desenvolvidos e novos contextos surgem. Fusão de dados é, em resumo, um conjunto de técnicas voltadas à integração de dados coletados por múltiplos sensores. Por esta definição, o tratamento de dados obtidos de tornazeleiras eletrônicas para monitoramento de indivíduos, está incluído dentro do conceito de Fusão de Dados.

Ainda como trabalho futuro, vale estudos acerca de algumas técnicas de Fusão de Dados que poderão ser aplicadas na análise preditiva baseando-se em estudos/observações do deslocamento dos pontos com o propósito de estimar encontros (rota de colisão entre dois pontos) ou ainda a indicação antecipada de infração de perímetro de zonas de inclusão ou exclusão.

# Capítulo 3

## Estado da Arte

Neste capítulo, abordamos os principais trabalhos cujos temas contenham alguma similaridade ao que foi proposto neste trabalho, seja com relação aos aspectos de funcionalidade, plataforma de desenvolvimento ou solução tecnológica.

Em seu trabalho, [Jakkhupan and Klaypaksee 2014] descreve um sistema de registro de eventos criminosos através de aplicação *web* com foco em dispositivos móveis (*smartphones* ou *tablets*). O registro é realizado por cidadãos onde a localização atual obtida pelo dispositivo é utilizada para identificação da área geográfica onde ocorreu o evento. Propõe como trabalho futuro a aplicação de técnicas de Mineração de Dados (*Data Mining*) sobre os registros como forma de estabelecer ações preventivas contra crimes. Uma possível integração deste sistema com o monitoramento por tornezeleiras proposto aqui poderia agilizar investigações através do cruzamento de dados e detecção de suspeitos presentes na área no momento do crime.

Na mesma linha de pensamento de Jakkhupan está [Oduor et al. 2014], que descreve em seu artigo uma aplicação *mobile* para registro de ocorrência criminosa em um Livro de Ocorrências (*Occurrence Book – OB*) suprimindo a necessidade do cidadão ir até uma agência de polícia para preencher formulários burocráticos. Desta forma, além de elevar o número de ocorrências registradas (muitas não são registradas por conta da burocracia), também reduziria possíveis erros no preenchimento como a indicação correta do local da ocorrência. Da mesma forma proposta por Jakkhupan, o sistema de monitoramento de tornezeleiras poderia ser complementado pelos dados coletados pelo sistema tornando o resultado mais detalhado.

Há uma iniciativa brasileira com a mesma iniciativa chamada "Onde fui roubado"([OndeFuiRoubado 2016]), cujo o acesso está disponível no sítio [www.ondefuiroubado.com.br](http://www.ondefuiroubado.com.br). Basicamente a vítima de algum roubo, furto, ou outro crime entra no site e registra onde e quando o delito ocorreu gerando uma base centralizada com estatísticas de regiões mais perigosas por estado ou cidade.

Os três trabalhos descritos acima buscam coletar dados através da contribuição de ci-



dados no registro de ações criminosas produzindo como resultado uma base de dados, geralmente disposta em mapas em aplicações web, com geolocalização de áreas com mais ou com menos ocorrências. Embora estas propostas utilizem recursos de geoprocessamento para combate ou auxílio na prevenção de ações criminosas, como é o caso do problema descrito neste trabalho, têm como objetivo principal prover a sociedade no geral com informações acerca de que regiões em sua cidade devem ser evitadas.

Porém, os agrupamentos de monitorados produzidos como solução deste trabalho poderiam ser complementados com os dados gerados pelos sistemas listados acima. Cruzando os locais, datas e horários dos eventos com os agrupamentos de indivíduos torna-se possível gerar informações mais detalhadas para subsidiar a polícia em investigações. Deve-se atentar, no entanto, qualquer julgamento prévio baseado apenas no cruzamento desses dados, preservando o direito de defesa dos eventuais suspeitos.

O trabalho de [Morreale and Suleski 2009] descreve um projeto para um Sistema de Identificação e Informação de Rede de Sensores Sem Fio (*Wireless Network Sensor Information and Identification System - WiNS Id*) onde um grande volume de dados temporais são coletados de sensores distribuídos geograficamente e armazenados em uma base de dados MySQL. A situação enviada pelos sensores é então mostrada numa interface gráfica.

Não se trata de um sistema voltado para prevenção de ações criminosas, mas aborda uma arquitetura de sensores distribuídos semelhantes ao encontrado no contexto de monitoramento por tornozeleiras eletrônicas. Diferente do trabalho proposto aqui, [Morreale and Suleski 2009] se mantém no conceito da arquitetura apresentando como resultado um gráfico de linhas com indicação das medições obtidas pelos sensores. Este trabalho objetiva uma visão mais prática da aplicação de focando *WiNS Id* buscando o apoio na redução de ações criminosas cometidas por grupos de indivíduos monitorados.

Usando técnicas de mineração de dados, [Sathyadevan et al. 2014] propõe uma abordagem para extração de dados para predição de crimes por áreas geográficas. O processamento desses dados segue um fluxo com as etapas Coleção de Dados, Classificação, Identificação de Padrões, Predição e Visualização. Dentre as fontes utilizadas como origem dos dados, cita *sites* em geral, *sites* de notícias, *blogs*, redes sociais, *feeds RSS*, etc. Propõe que os dados não estruturados sejam armazenados em um SGBD *NoSQL* denominado *Mongo DB*.

A implantação do projeto abordado por [Sathyadevan et al. 2014] implica na implantação de uma plataforma onde os dados seriam gerados por técnica de *big data*. A identificação da localização das ocorrências detectadas seria possível porém menos precisas que a solução abordada neste trabalho. Além disso, haveria dificuldade também na identificação dos componentes dos grupos. De qualquer forma, um futuro estudo acerca de uma possível integração dos dados produzidos pelo projeto de [Sathyadevan et al. 2014] com os agrupamentos detectados neste projeto pode tornar-se relevante, tendo em vista o potencial e a abrangência inerentes aos projetos de *big data*.

Em [Urbano and Dettki 2014] são tratadas questões acerca da criação e manutenção de

uma base de dados em PostgreSQL com a extensão PostGIS, armazenando dados geográficos transmitidos por sensores localizados na Itália. Descreve as etapas de criação da base de dados e as tabelas necessárias para demonstração de uma aplicação com armazenamento de dados geográficos no SGBD. O trabalho de Urbano apoia uma solução de plataforma para armazenar e processar dados geográficos, servindo como guia para uma das possíveis soluções a ser adotada neste trabalho.

No trabalho de [Urbano and Dettki 2014] são tratadas apenas conceitos, implementação e carga de uma pequena base geográfica na plataforma PostgreSQL suportada pela biblioteca PostGIS. Este trabalho vai além, abordando o processamento geográfico sobre uma base geográfica já existente (pontos coletados na Central de Monitoramento) para extração e fornecimento de dados.

Outros trabalhos apresentam assuntos relativos à aplicação da tecnologia de GPS para monitoramento, como por exemplo o de [Ki and Hee 2011] que propõem uma arquitetura de monitoramento por torneleiras com uma topologia com agentes intermediários autônomos entre os dispositivos móveis e a central. Os agentes são componentes de software dinâmicos que fornecem serviços de operação colaborativa. Utilizando estes agentes o sistema poderia tomar decisões descentralizadas agilizando o processo de alertas. Cita como exemplo os diversos níveis de alerta sobre a proximidade de um criminoso sexual e crianças monitoradas.

A solução de [Ki and Hee 2011] poderia ser aplicada em substituição ao atual sistema de coleta de pontos. Vale ressaltar que, por apresentar uma estrutura descentralizada, dificultaria ações que exigem o processamento de forma centralizada. A descentralização apresentada como vantagem, pode apresentar dificuldades relativas à arquitetura, como a necessidade de uma maior tolerância a falhas de rede, uma das características essenciais do monitoramento de indivíduos.

[Carlino et al. 2013] argumentam sobre a influência da proximidade física de laboratórios de pesquisa e desenvolvimento (P&D) na repercussão do conhecimento na área de concentração dos mesmos. Para tanto, compara a localização dos laboratórios no território dos EUA com o registro de patentes na mesma área, evidenciando sua ligação. Então, aborda uma maneira de medir a extensão da concentração espacial das atividades dos laboratórios e definir o agrupamento formado pelos laboratórios próximos, considerando um círculo em volta de cada ponto de localização, com um raio inicial de um quarto de milha. Em seguida, enumera a quantidade de pontos dentro do círculo. Como resultado, muitos círculos se sobrepõem formando o agrupamento a ser analisado e comparado com o registro de patentes.

No problema apresentado no trabalho de [Carlino et al. 2013], não há necessidade de se rastrear o agrupamento acompanhando, por exemplo, sua duração e quantidade de elementos. É apresentado apenas um conceito e não um algoritmo para detecção do agrupamento.

A seguir, são apresentados alguns artigos que tratam de algoritmos para processamento geográfico de pontos e a detecção de *clusters*. No geral, baseiam-se na procura de concentração de pontos analisando sua distribuição no espaço. Alguns não levam em conta

a necessidade específica de identificar agrupamentos dentro de uma distância mínima que caracterize uma reunião de forma efetiva.

[Liu et al. 2012] abordam algoritmos que identificam agrupamento de objetos classificados em categorias, considerando aspectos puramente geográficos ou outros atributos associados. Aborda principalmente o algoritmo “*density-based spatial clustering - DBSC*” que detecta agrupamentos usando tanto a proximidade espacial quanto a similaridade de atributos. O DBSC envolve a construção de relacionamentos de proximidades entre pontos obtidos por meio da triangulação de *Delaunay* ([Chew 1989]).

Para obter os triângulos formados por pontos próximos necessários ao algoritmo, a distância entre os pontos deve ser previamente calculada. Este aspecto pode comprometer a restrição de tempo de processamento, requisito relevante no caso deste trabalho.

Já o algoritmo DBSCAN [Louhichi et al. 2014] busca a identificação de agrupamentos com diversos tipos de objetos geográficos (pontos, polígonos, linhas, etc.). Cada grupo adjacente em um dado raio contém pelo menos um número mínimo de pontos, ou seja, sua densidade extrapola algum limite, o que deixa claro que o processamento é executado ponto a ponto utilizando a relação de distância entre pontos. O artigo propõe estimar o valor da distância para diferenciar a ideia de concentração de pontos da ideia de pontos dispersos, fora da concentração eliminando ruídos.

Quando comparado ao problema deste trabalho, o valor estimado não é necessário uma vez que já o estabelecemos em função da precisão (*accuracy*) do GPS, através de uma distância mínima a ser considerada. Além disso, o algoritmo proposto por [Louhichi et al. 2014], quando comparado ao proposto neste trabalho, possui certa complexidade para tratar os tipos diferentes de objetos analisados. Tal complexidade não existe no problema abordado neste trabalho uma vez que são tratados apenas pontos.

Os trabalhos de [Liu et al. 2012] e [Louhichi et al. 2014], até por estarem desvinculados da questão dos LETS, não atendem a requisitos do problema abordado no presente artigo, ou seja, não consideram a evolução do agrupamento (análise temporal) considerando a duração da concentração de pontos e seu tamanho (quantidade de elementos).

Dada a necessidade de processar os pontos geográficos dentro de uma janela de tempo específica, mesmo considerando uma grande quantidade de coordenadas geográficas na amostra coletada, torna-se importante a adoção ou elaboração de algoritmos paralelizáveis, principalmente no que se refere a detecção de pares de pontos próximos. O trabalho de [Ding and Densham 1996]. apresenta uma abordagem de “decomposição do domínio” possibilitando a paralelização do algoritmo permitindo melhorar, se for o caso, o tempo de execução do processamento.

Para o problema tratado neste trabalho, no que se refere ao aspecto de desempenho, utilizou-se uma versão adaptada da proposta de [Ding and Densham 1996]. Enquanto eles propõem diversas formas de subdivisão de áreas, nesta versão adaptada o algoritmo foi simplificado optando por divisões recursivas em quadrantes.

[Ding and Densham 1996] apresentam um trabalho abrangente sobre as possíveis estratégias para paralelização de modelagem espacial, sobre as quais se baseia o trabalho aqui proposto, principalmente no que se refere ao ganho de desempenho necessário para atender ao requisito de tempo de execução dentro de uma janela limite. Parte dos conceitos apresentados por eles foram abordados anteriormente neste capítulo. Todavia, vale explorar aqui mais alguns conceitos relevantes para o entendimento do problema a apresentação de uma solução.

### **3.1 Particionamento simples para decomposição de domínios espaciais**

A correta identificação da estrutura do problema é importante para o projeto do algoritmo paralelo. Em alguns casos de problemas espaciais a decomposição do domínio, ou seja o recorte da área a ser analisada, é vantajosa. A decomposição do domínio em quadrados e retângulos tem vantagem por ter sua implementação mais simples e pela natureza do sistema de coordenadas espaciais.

A decomposição pode se dar em uma dimensão, horizontal ou vertical, ou em duas dimensões com a combinação das duas. A decomposição do domínio em células de formato e tamanho iguais facilita a distribuição e a montagem final, sendo que este último pode se dar pela mesclagem horizontal ou vertical. Outros tipos de decomposição, como por exemplo baseado em paralelogramas, losangos ou triângulos são eventualmente úteis em casos bem específicos de transformações ou projeções de sistemas de coordenadas.

Divisão uniforme de domínios espaciais irregulares, como por exemplo a dispersão irregular de pontos no domínio espacial, provocam uma distribuição de carga sem balanceamento, ou seja, alguns nós trabalharão mais que outros. Para buscar um melhor balanceamento da carga é necessária a divisão do domínio em áreas diferentes acompanhando a irregularidade do espaço.

Uma maneira de particionamento mais eficiente para uma distribuição irregular do espaço é a divisão da área em células de tamanhos iguais e então distorcer os retângulos movendo os pontos de rede para equalizar as partições (Figura 3.1). O que precisa ser observado nesta solução é o custo computacional necessário para distorcer os retângulos.

#### **3.1.1 Subdivisões recursivas para decomposição de domínios espaciais**

Subdivisões recursivas baseadas em *Quadtree* geram partições (subáreas) retangulares a partir de uma amostra de espaço também retangular. Várias pesquisas abordam essa ideia de divisão do espaço em retângulos. Estas técnicas, denominadas Tesselação Recursiva Adaptativa (*adaptive recursive tessellation - ART*), podem ser representadas por dados espaciais

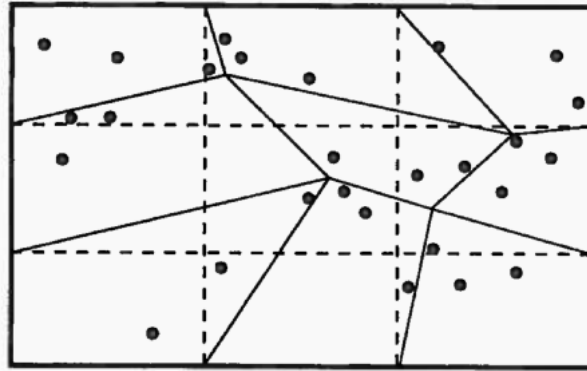


Figura 3.1: Particionamento de rede alternado. Fonte: [Ding et al. 1992]

hierárquicos ou por mapas.

Para resolver o problema de balanceamento em domínios irregulares pela divisão uniforme do espaço, um método a ser adotado é a Bisseção Alternada Recursiva Dinâmica (*dynamic recursive alternating bisection - DRAB*) que usa a partição de objetos ou elementos distribuídos desigualmente. Primeiramente uma linha é usada para dividir o espaço em duas partições com número de elementos semelhantes. Em seguida, cada partição é dividida recursivamente em mais duas subpartições por linhas perpendiculares à anterior (Figura 3.2). A divisão deve ir até o ponto em que o número de elementos esteja adequado ao esforço de processamento. O resultado final, apesar de apresentar tamanhos e formatos das células de formato irregular, tende a manter o balanceamento mais equilibrado em função da quantidade regular de elementos.



Figura 3.2: Bisseção Alternativa Recursiva Dinâmica. Fonte: [Ding et al. 1992]

### 3.2 Sobreposição espacial versus decomposições sem sobreposição

A decomposição do domínio espacial, seja pelo particionamento simples, seja pela divisão recursiva, divide a área espacial em um número específico de partes menores que, apesar de cobrir totalmente a área original, não possuem qualquer sobreposição entre áreas vizinhas.

Para problemas de processamento de escopo estritamente local, a ausência de

sobreposição é a forma mais natural e eficiente de paralelizar o processamento, pois o resultado pode ser simplesmente montado ao final através da concatenação dos resultados parciais.

Entretanto, em outros tipos de problemas há a dependência de dados de uma subárea para outra que pode ser resolvido ou por intercomunicação entre os processos ou pela sobreposição de áreas. Sem isso, a montagem do resultado final fica comprometido podendo apresentar resultados incorretos.

Como vimos anteriormente, a dependência sobre comunicação pode degradar significativamente o desempenho do algoritmo em função da quantidade de dados e frequência de comunicação. Portanto, a solução mais adequada nestes casos é a extensão da área de cada subdomínio no sentido horizontal e/ou vertical conforme a dependência (Figura 3.2).

Uma porção pequena de dados sobrepostos evita a necessidade de comunicação entre os processos permitindo o ganho de desempenho conforme o aumento da granularidade. Porém, em caso de escopo global onde a dependência de dados é relativamente grade provocando uma extensão muito grade das subáreas, este método pode se tornar ineficiente por conta do processamento adicional na montagem do resultado final.

Após a decomposição do domínio, o processamento de cada subárea é submetida à um processador (ou nó). Em virtude da quantidade de peças a serem processadas e a disponibilidade de processadores, alguns processamentos podem ficar enfileirados aguardando sua vez.

Para atender ao algoritmo de detecção de pares de pontos próximos utilizou-se a opção de sobreposição espacial proposta por [Ding and Densham 1996]. Porém, esta sobreposição é parcial, avançando o suficiente apenas para que o algoritmo alcance os pontos próximos em áreas vizinhas. Desta forma, não há a necessidade de comunicação entre os processamentos de subáreas dispensando este tratamento.

# Capítulo 4

## Definição do Problema e Proposta de Solução

O sistema de rastreamento por dispositivos via satélite, consiste no agrupamento de diversas tecnologias para rastrear sentenciados em regime aberto, semi-aberto e em prisão domiciliar. Integrado às ações conjuntas das polícias civis e militares, permite uma fiscalização mais eficiente através da Central de Monitoramento, que transmite as ocorrências aos postos policiais mais próximos da localidade onde é constatada qualquer irregularidade através do uso do dispositivo.

Várias empresas, nacionais e estrangeiras, têm oferecido soluções de monitoramento eletrônico através de tornozeleiras no Brasil. Como funcionalidade básica, utiliza a geolocalização através de GPS no equipamento e o envio de pontos através da rede telefônica celular detectando violações de zonas de inclusão (áreas que o monitorado não pode sair) ou exclusão (áreas que o monitorado não pode adentrar). A central de monitoramento é responsável por processar os pontos e enviar alertas às autoridades responsáveis em caso destas violações.

### 4.1 Problema

O acionamento das autoridades através de alertas é gerado após a ocorrência da identificação da violação tornando o sistema reativo com relação a sequência de ações, muitas vezes tardia. Eventuais ações criminosas podem ser potencialmente detectadas antes que aconteçam. Para tanto, existe conjuntos de pontos no formato de coordenadas geográficas que são coletados periodicamente e podem ser coletivamente processados produzindo informações relevantes para subsidiar análises de riscos.

Uma forma de produzir informações a partir dos pontos coletados e armazenados na Central de Monitoramento é o processamento para detecção de agrupamentos indicando eventuais reuniões de elementos monitorados. Tais agrupamentos, acompanhados de outros

atributos relevantes como a duração do encontro, quantidade de elementos, frequência de encontro, horários comuns e localização, podem embasar eventuais investigações e análise de potenciais riscos relativos á ações criminosas iminentes.

Antes de apresentarmos os algoritmos necessários para produção dos resultados esperados, algumas definições iniciais devem ser estabelecidas para entendimento dos requisitos acerca da detecção de agrupamentos. A primeira a ser abordada é a definição de proximidade de forma a filtrar os agrupamentos que de fato demonstram que alguns monitorados estão realmente reunidos. Outra definição é o tempo mínimo de duração de um grupo para assumirmos que realmente trata-se de uma reunião ou de apenas um encontro casual.

Também é abordada a periodicidade da coleta dos pontos que impõe uma janela de tempo definindo um prazo a ser cumprido para evitar o atrasos acumulativos no processamento da amostra. A complexidade computacional deve ser avaliada para manter o desempenho da execução de todas as etapas dentro da janela. Uma das alternativas de solução é a distribuição do processamento para permitir a paralelização.

#### 4.1.1 Definição de Proximidade

Na ótica do monitoramento de pessoas para aplicação da lei, a concentração de pontos em uma área geográfica não necessariamente indica um encontro ou reunião de monitorados. Para que dois ou mais monitorados possam ser considerados reunidos, deve haver uma distância mínima a ser estabelecida entre suas coordenadas.

Para a definição de tal distância, levamos em consideração a margem de erro ( $\epsilon$ ) inerente aos equipamentos de geolocalização com GPS (Figura 4.1).

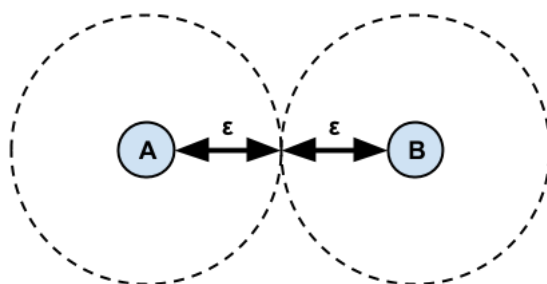


Figura 4.1: Distância mínima entre dois pontos para definição de proximidade.

Segundo o relatório da [FAA 2014], a precisão dos dispositivos GPS acompanhados naquele trabalho gira em torno de 10 metros. Considerando este erro de precisão em cada um dos elementos da comparação, estabelecemos um parâmetro limiar para esta definição em 20 metros. Ou seja, para efeito de filtro de pares de pontos próximos, dois pontos quaisquer separados por distância inferior ou igual a 20 metros serão considerados monitorados próximos.

Considerando grandes distâncias, a precisão pode ser influenciada pela curvatura da terra



sendo recomendada nestes casos, a aplicação de um sistema de referência adequado (*SRID*) conforme abordado na sessão 2.1.3. Porém, devido a curta distância referenciada em nosso problema (menor ou igual a 20 metros), podemos aplicar o cálculo da distância Euclidiana, onde os pontos medidos estão dispostos em uma superfície plana.

A evolução da tecnologia dos dispositivos GPS vem ganhando força nos últimos anos melhorando continuamente sua precisão no posicionamento geográfico. Embora tenha sido inicialmente estabelecido o parâmetro de 20 metros inicial, este valor pode sofrer eventuais ajustes no futuro.

#### **4.1.2 Duração mínima de encontros para definição de reunião**

Agrupamentos cuja a duração seja muito pequena, podem indicar que o encontro foi apenas uma aproximação casual e não um encontro de fato. Quando dois elementos se cruzam em uma avenida, ou entram por coincidência em um recito público, são situações de aproximação que embora detectadas pelo sistema, deveriam ser descartados. Desta forma, deve-se descartar os grupos cujo tempo mínimo seja menor que um valor previamente estabelecido.

Sabe-se se um grupo continua reunido quando este é detectado no processamento do último realizado sobre os pontos coletados. Os grupos detectados em um processamento sevem ser comparado ao histórico dos grupos detectados anteriores atualizando a duração e a quantidade de elementos do agrupamento.

Outra restrição a saber é a precisão dos horários que um encontro se inicia e termina. Entre uma coleta de posição geográfica e a próxima, um dispositivo pode se deslocar para longe ou para perto de outro dispositivo. Não é possível determinar precisamente em que momento um par de pontos entrou na margem de distância mínima. O mesmo se aplica quando dois dispositivos se afastam. Desta forma, os horários de início e de término dos encontros serão registrados com base nos momentos definidos das coletas de pontos, e assim, a precisão dependerá do período de coleta adotado (janela de tempo de processamento).

#### **4.1.3 Execução dos algoritmos dentro do tempo definido**

A detecção de agrupamentos ocorrida em uma determinada coleção de pontos recebidos no período deve ocorrer antes do envio do próximo lote de pontos. A janela deve ser fixada inicialmente, antes de se iniciar o processamento, e seu tamanho impacta as necessidades de desempenho dos algoritmos, pois caso o prazo seja ultrapassado, há o risco de acúmulo com o processamento sobrecarregando os recursos computacionais, e em última instância, parada do sistema por completo.

Esta preocupação é agravada pelo aumento da complexidade computacional na etapa de detecção de pontos próximos em função da quantidade de pontos a serem comparados para o

cálculo da distância. À medida que aumenta a quantidade de pontos pertencente ao conjunto coletado, a quantidade de comparações necessárias para detecção de pares próximos também aumenta. A quantidade de comparações  $C_{n,p}$  é dada pela fórmula da análise combinatória por combinação simples. A Figura 4.2 mostra o crescimento de comparações necessárias conforme aumenta-se a quantidade de pontos.

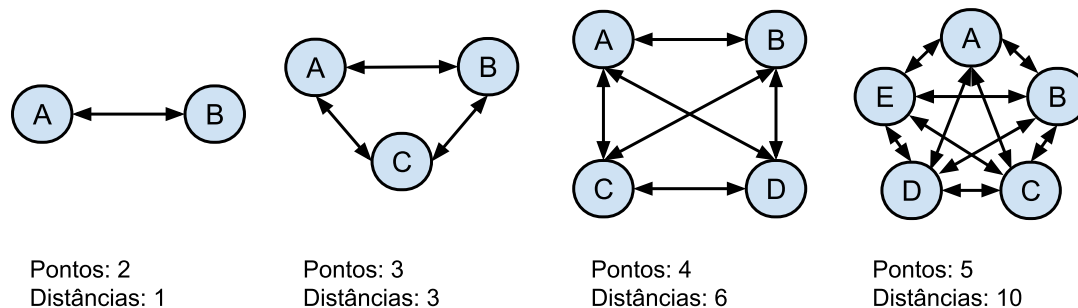


Figura 4.2: Aumento do cálculo de distâncias em função do aumento da quantidade de pontos.

Em uma amostra com 10.000 pontos, por exemplo, teríamos aproximadamente 50 milhões de cálculos de distâncias. Por conta dessa quantidade de cálculos a serem executados em uma janela de processamento limitada, este tópico representa um dos pontos críticos para implantação de uma solução. Lembrando que a quantidade de elementos monitorados pode crescer com a evolução na implantação do sistema, dada o déficit de vagas e o aumento da população carcerária que em 2013 era de 574 mil (Figura 1.2).

#### 4.1.4 Divisão do espaço em subáreas para permitir paralelização

A divisão de problemas com intuito de permitir soluções paralelizadas apresenta diversos desafios pois depende da natureza do problema, das características dos dados, do algoritmo escolhido, etc. Conforme revisão literária, [Ding and Densham 1996] aborda vários problemas e soluções acerca da paralelização no geoprocessamento. Dentre as alternativas discutidas, a de subdivisões recursivas para decomposição de domínios espaciais (Sessão 3.1.1) é uma das alternativas viáveis para detecção de pontos próximos em monitoramento por dispositivos eletrônicos.

Desta forma, aplica-se a divisão do espaço total inicial em subáreas de forma que os diversos nós ou processadores poderão manipular várias áreas de forma concomitante. Como a natureza de monitoramento de pessoas por tornozeleiras eletrônicas possui distribuição dos pontos sobre a área de forma extremamente irregular, é necessário aplicar alguma estratégia para um balanceamento da carga que aproveite bem os recursos de processadores disponíveis. Em resumo, a distribuição irregular de pontos impõe uma divisão também irregular da área para equilibrar a distribuição da carga.

Para definição da área inicial no problema, considerou-se um retângulo contendo todos

os pontos da amostra, gerado a partir dos pontos mais afastados do centro.

#### **4.1.5 Sobreposição espacial para resolver dependências**

A proposta de subdivisão da área para permitir a paralelização apresenta alguns efeitos colaterais relacionados às necessidades da modelagem espacial. A comparação de distância entre pontos cria uma interdependência entre as subáreas, dado que um ponto pode estar próximo de outro mas em áreas distintas, porém vizinhas.

Para resolver esta interdependência, há dois caminhos possíveis: i) estabelecer canais de comunicação entre os processos que tratam cada subárea e ii) alterar a subdivisão das áreas extrapolando seus limites sobrepondo as subáreas adjacentes.

A primeira alternativa, estabelecer canais de comunicação, traz alguns desafios: i) complexidade para a implementação; ii) interdependência entre os processos; e iii) forte dependência da interconexão de rede.

A segunda alternativa, expandir a subáreas sobrepondo as áreas vizinhas, acompanha o que foi abordado na Sessão 2.4.4.2, mostrando-se alinhado com o tipo de problema proposto neste trabalho. Esta solução permite uma situação em que todas as áreas podem ser processadas de forma independente mesclando os resultados parciais para produzir o resultado final.

#### **4.1.6 Identificação do mesmo agrupamento em processamentos subsequentes**

Para estabelecer a continuidade de um grupo através do tempo, é necessário verificar sua detecção durante uma série de processamentos consecutivos. Logo a duração do encontro parte do momento do primeiro processamento em que foi detectado até o momento da última vez que foi detectado.

Um desafio adicional ocorre quando um grupo tem alguns de seus membros afastados ou houver a adição de novos elementos durante sua existência. Isto impõe à solução proposta a necessidade de permitir flexibilidade na comparação do grupo detectado com os grupos vigentes no histórico, com relação ao número de elementos em comuns.

#### **4.1.7 Quantidade de Elementos no Agrupamento Detectado**

Para os agrupamentos detectados, sua quantidade de elementos é um dado relevante para a análise de risco. Grupos com 10 elementos podem representar um risco maior do que grupos com 2 elementos, uma vez que isto pode representar um delito de maior proporção e de maneira mais organizada através da divisão de atividades entre os membros do grupo.

### **4.1.8 Freqüência de encontro dos agrupamentos**

Um mesmo agrupamento pode ser detectado várias vezes ao dia ou em um determinado período de dias. A quantidade de vezes que um mesmo grupo se encontra em determinado período de tempo pode influenciar seu fator de risco, pois implica em certa probabilidade do grupo voltar a se encontrar. Logo, o fornecimento desse dado é um importante item a ser considerado no processamento dos pontos.

### **4.1.9 Horários comuns aos encontros**

Alguns grupos podem se encontrar em horários aleatórios enquanto em horários regulares. Para este último caso é relevante relatar o horário comum para encontro daquele grupo.

É necessário levantar além do horário comum de encontro, a relação de encontros dentro daquele horário para estabelecer a freqüência e uma probabilidade de que grupo voltaria a se encontrar novamente.

### **4.1.10 Área geográfica comuns aos encontros**

Alguns grupos podem se encontrar costumeiramente em um local. A identificação deste local é uma informação relevante para diligências e investigações. O que deve ser observado neste caso é que um grupo pode estar parado em um determinado instante ou em deslocamento (Figura 4.11). Quando um grupo está em deslocamento a definição de área comum para os encontros não faz sentido, embora durante o deslocamento o grupo possa eventualmente cruzar um mesmo local percorrido anteriormente tornando aquela área mais comum.

Uma aplicação prática seria subsidiar investigações mais detalhadas sobre o comportamento do determinados grupos. Através de análises adequadas, o horário e o local do próximo encontro poderão ser inferidos apoiando equipes de investigação ou vigilância.

## **4.2 Modelo Sistêmico e Proposta de Solução**

O conjunto formado pelos dispositivos eletrônicos de monitoramento representado pelas tornozeleiras eletrônicas pode ser visto como uma Rede de Sensores Sem Fio, pois cada elemento possui dimensões reduzidas (portátil), possui certa capacidade de processamento, recursos de comunicação (GSM/GPRS) e são dotados de algum tipo de sensor, um circuito GPS neste caso (Sessão 2.5).

No caso de monitoramento por tornozeleiras, a comunicação se dá diretamente com a Central de Monitoramento pelos protocolos GSM/GPRS. Esta característica, entre outros aspectos, elimina a interdependência entre os dispositivos com relação a comunicação, mas

por outro lado, exige a cobertura da rede de celulares na região monitorada. Para reduzir a indisponibilidade por ausência de sinal de uma operadora, alguns dispositivos suportam a conexão com duas ou mais operadoras.

Ainda no contexto de Redes de Sensores Sem Fio, as tecnologias de Fusão de Dados devem ser consideradas para a integração dos dados enviados pelos sensores (Sessão 2.6.1).

A solução proposta foi construída observando os conceitos de Rede de Sensores Sem Fio e da Fusão de Dados, além de princípios de algoritmos surgidos anteriormente a estes termos, como as técnicas abordadas no trabalho de [Ding and Densham 1996], [Louhichi et al. 2014] e outros listados na Sessão 3.

A Figura 4.3 apresenta o modelo proposto neste trabalho que divide o processamento em três etapas: 1) Recebem da rede os pontos enviados pelos dispositivos e calcula os pontos próximos (Detecção de Pares); 2) Agrupam os pontos próximos (pares) em grupos (Detecção de Grupos); e 3) Acrescentam a duração do grupo e a quantidade de participantes e prepara os dados para consultas complementares (Detecção de Riscos).

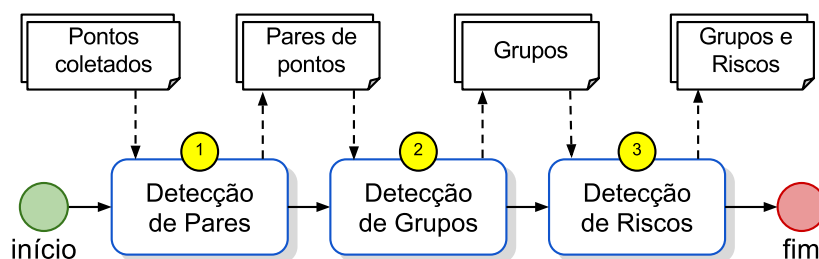


Figura 4.3: Etapas do processamento.

Cada etapa recebe uma coleção de dados de entrada e produz uma saída que é, nos casos da etapa 1 e 2, entrada para as etapas subsequentes. Tais entradas e saídas são conjuntos estruturados de dados cujos leiautes são detalhados neste capítulo.

Os dados de entrada da etapa 1 (Detecção de Pares) são os pontos geográficos recebidos e armazenados na Central de Monitoramento. O mecanismo de coleta destes dados já está implantado no sistema atual para detecção de violação de zonas de inclusão ou exclusão. Desta forma, não abordaremos aqui os detalhes acerca do envio, tratamento e armazenamento destes dados trafegados desde o dispositivo físico até a Central de Monitoramento.

Também não são considerados os eventuais pontos inválidos ou atrasados, pois assumiu-se que tal tratamento foi previamente realizado pelo processo de verificação de violação de zonas.

As etapas 1 e 2, Detecção de Pares e Grupos respectivamente, são representados como um processo unificado em alguns algoritmos pesquisados. Porém, devido à criticidade no desempenho do algoritmo 1 para atendimento da janela temporal, tais algoritmos foram abordados de forma separada.

O resultado final entregue pela etapa 3 (Grupos e Riscos) é uma coleção histórica com-

posta de grupos e seus dados complementares como duração e quantidade de elementos. Outros dados como a frequência e os horários e regiões comuns aos encontros são obtidos posteriormente através de consultas/processamentos sobre a saída da etapa 3.

Um dos atributos tratados, a identificação do dispositivo (tornozeleira), corresponde a um número serial vinculado ao hardware. A correlação entre este número e o cadastro individual do monitorado é tratado em sistema específico. Desta forma, a manipulação dos dados cadastrais dos monitorados não faz parte do escopo deste trabalho.

#### 4.2.1 Etapa 1 - Detecção de Pares

A primeira etapa recebe como entrada uma coleção de pontos recebidos pela Central de Monitoramento ("Pontos coletados" na Figura 4.3) em períodos regulares de tempo. A estrutura destes dados está detalhada na Tabela 4.1.

Atributo	Tipo	Descrição
Dispositivo ID	Número Inteiro	Número identificador do dispositivo.
Data/Hora	Data Hora	Data, hora e minuto que o ponto foi coletado.
Ponto	Coordenada Geográfica	Ponto geográfico composto pela Latitude e Longitude.

Tabela 4.1: Pontos coletados – Entrada do algoritmo 1 (Detecção de Pares).

Como resultado da etapa 1, uma coleção de pares de pontos próximos ("Pares de pontos" na Figura 4.3) deve ser produzindo seguindo o leiaute definido na Tabela 4.2.

Atributo	Tipo	Descrição
Data/Hora	Data Hora	Data, hora e minuto da coleta dos pontos.
Dispositivo A	Número Inteiro	Número identificador do dispositivo A.
Ponto A	Coordenada Geográfica	Ponto geográfico composto pela Latitude e Longitude
Dispositivo B	Número Inteiro	Número identificador do dispositivo B.
Ponto B	Coordenada Geográfica	Ponto geográfico composto pela Latitude e Longitude

Tabela 4.2: Estrutura de saída de dados da Etapa 1 (Detecção de Pares).

Os algoritmos necessários para executar o que for proposto na etapa 1 têm como objetivo comparar cada ponto com os demais do conjunto gerando uma lista daqueles pares de pontos cuja distância seja menor que o parâmetro, inicialmente definido em 20 metros conforme discutido na Sessão 4.1.1.

Esta etapa representa, em termos computacionais, umas das mais onerosas, pois implica na combinação entre todos os pontos seguido do cálculo da distância e filtro conforme discutido na Sessão 4.1.3.

Antecipando parte da solução de problema de desempenho desta etapa, foram elaborados dois algoritmos, sendo o primeiro para dividir o espaço total em subáreas e o segundo para realizar a comparação entre os pontos identificando os pares de pontos próximos.

#### 4.2.2 Algoritmo 1.1 – Divisão do espaço geográfico em subáreas

Este algoritmo inicialmente gera uma área geográfica retangular que abrange todos os pontos da amostra. Em seguida, divide esta área em quatro subáreas de igual tamanho (quadrantes). As subáreas que apresentam quantidade de pontos maior que o limite estabelecido em um certo parâmetro, são re-submetidos para nova subdivisão. Este processo se repete até que todas as subáreas tenham, no máximo, a quantidade definida pelo parâmetro (Figura 4.4). Este método é uma variante das alternativas propostas por [Ding and Densham 1996] e por [Xia et al. 2012] através de *quadtree*.

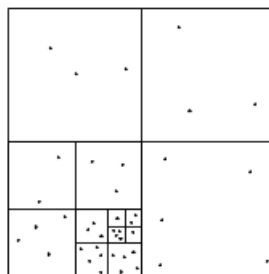


Figura 4.4: Subdivisão da área total em subáreas - Fonte: adaptado de [Ding and Densham 1996]

O cálculo das distâncias ocorre apenas dentro das subáreas em que se encontram os pontos, reduzindo o esforço computacional. Porém, há situações em que dois pontos estarão próximos, mas em subáreas subjacentes. Como alternativa para resolver este problema, ao criar uma nova subárea expandimos sua abrangência adicionando uma margem equivalente à distância mínima para detecção de pontos próximos (Figura 4.5). Esta situação está detalhada na Sessão 4.1.5.

A partir desta expansão, processamento de subáreas distintas poderão identificar o mesmo par de pontos próximos. Tais duplicações serão eliminadas no algoritmo de Detecção de Grupos (Sessão 4.2.4) referente à etapa 3 do modelo sistêmico (Figura 4.3).

Os passos iniciais do Algoritmo 1.1 são mostrados na Figura 4.6.

A descrição de cada passo deste algoritmo está descrito na tabela 4.3.

O passo 3 do algoritmo 1.1 recebe como parâmetro uma área (ou subárea) e a divide em novas subáreas (quadrantes). Na medida em que os pontos ultrapassam a quantidade máxima estabelecida, esta função é chamada recursivamente até que nenhum dos quadrantes tenham mais pontos que o permitido.

A Figura 4.7 descreve os passos da função 3.

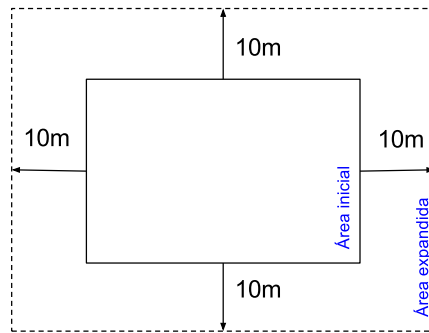


Figura 4.5: Expansão da área para contemplar pontos próximos em áreas adjacentes – Fonte: adaptado de [Ding and Densham 1996]

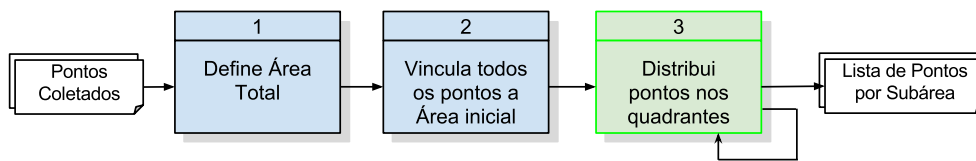


Figura 4.6: Algoritmo 1.1 – Distribuição dos pontos em subáreas.

#	Descrição
1	Gera uma área geográfica retangular que cobre todos os pontos da amostra a ser processada através dos pontos mais afastados do centro.
2	Registra todos os pontos na área inicial.
3	Função recursiva que divide a área recebida em quadrantes (4 novas subáreas) e reatribui os pontos vinculando aos novos quadrantes.

Tabela 4.3: Detalhamento dos passos do algoritmo 1.1 - Distribuição de Pontos em Subáreas.

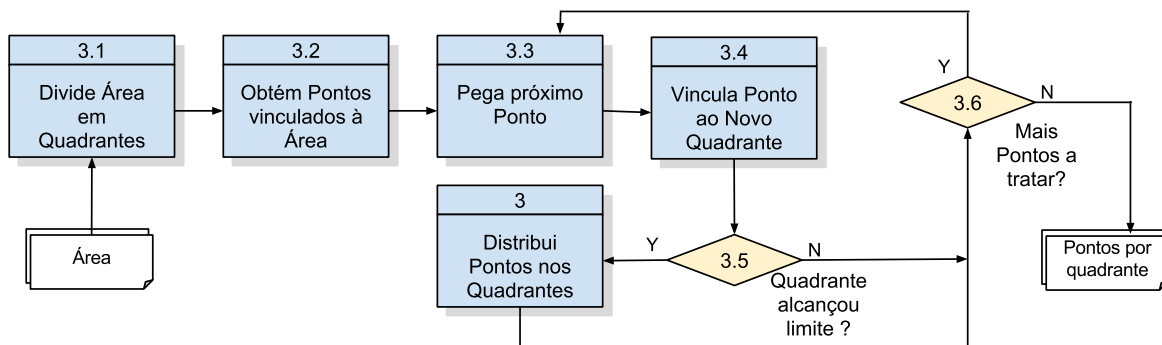


Figura 4.7: Detalhamento da função recursiva do algoritmo 1.1.

A Tabela 4.4 descreve os passos que compõem a função 3.

O algoritmo 1.1 gera uma lista de pontos com as respectivas subáreas conforme mostrado na Tabela 4.5. A saída do algoritmo 1.1 é uma lista dos pontos recebidos como entrada acrescido de um número identificado de área.



#	Descrição
3.1	Recebe uma área e a divide em quadrantes. Cada quadrante é uma nova subárea.
3.2	Varre os pontos vinculados à área fornecida através de um laço.
3.3	Pega o próximo ponto da varredura do item anterior.
3.4	Vincula o ponto ao novo quadrante de acordo com sua coordenada.
3.5	Se a quantidade de pontos vinculados ao novo quadrante alcançou o limite, a função “3 – Divide Área em Quadrante” é chamada recursivamente recebendo como parâmetro o novo quadrante como área.
3.6	Se ainda há pontos a serem tratados desvia o fluxo para tratar o próximo.

Tabela 4.4: Detalhamento dos passos da função recursiva.

Atributo	Tipo	Descrição
Dispositivo ID	Número Inteiro	Número identificador do dispositivo.
Data/Hora	Data Hora	Data, hora e minuto da coleta dos pontos.
Ponto	Coordenada Geográfica	Ponto geográfico composto pela Latitude e longitude.
Subárea	Número Inteiro	Número da subárea onde está o ponto.

Tabela 4.5: Estrutura de saída do algoritmo 1.2 (Detecção de Grupos).

### 4.2.3 Algoritmo 1.2 – Detecção de Pares em Subáreas

O algoritmo 1.2 pode trabalhar de duas formas: i) aplicado diretamente sobre a área geográfica com todos os pontos da amostra, sem considerar a subdivisão do algoritmo 1.1. Neste caso, trata-se da execução da etapa 1 sem paralelismo; e ii) aplicado de forma paralela sobre cada uma das subáreas resultantes do algoritmo 1.1.

A Figura 4.8 mostra os passos do algoritmo 1.2. Este algoritmo é composto de dois laços aninhados (Passos 1 e 3) que combinam todos os pontos e aplica filtro antes do cálculo da distância (Passo 5) e depois do cálculo (Passo 7). Este algoritmo segue a definição abordada na Sessão 4.1.3, onde 10.000 pontos geram aproximadamente 50 milhões de cálculos de distância.

A descrição dos passos que compõem o algoritmo 1.2 encontram-se na Tabela 4.6.

A Figura 4.9 mostra a execução paralelizada do algoritmo 1.2. O número de filas de processamento disponíveis (*threads*) não necessariamente acompanha a quantidade de subáreas a processar. A política de distribuição das subáreas nas filas (escalonamento) segue o método *round-robin*, onde cada fila recebe sequencialmente uma subárea e as demais que sobraram são redistribuídas seguindo a mesma ordem até que todas as subáreas estejam atribuídas a uma das filas.

O resultado do algoritmo de Detecção de Pares é uma coleção de pares de pontos próximos com a estrutura detalhada na Tabela 4.2. Em caso de processamento paralelo, todas as subáreas deverão ser processadas antes de iniciar a etapa 2 (Detecção de Grupos).

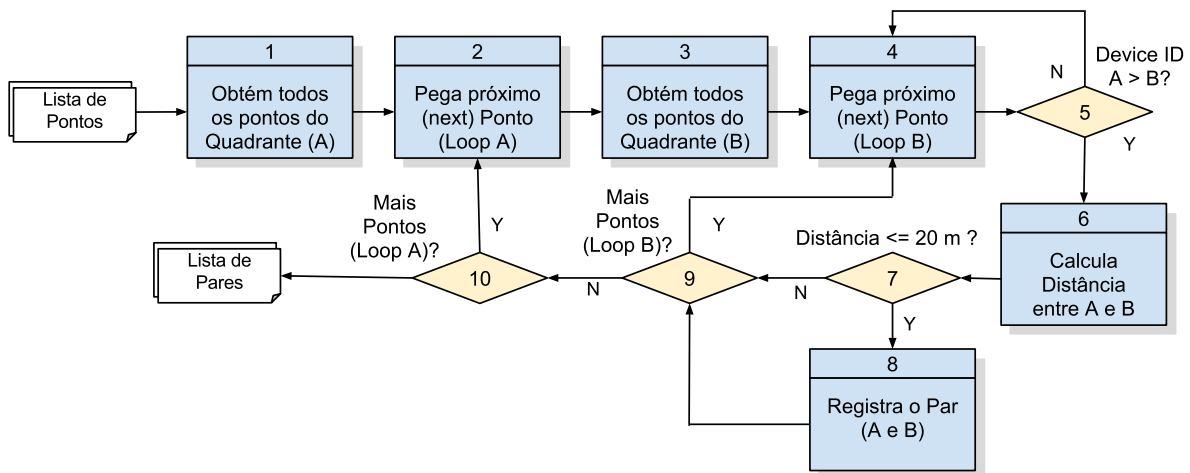


Figura 4.8: Algoritmo 1.2 - Detecção de Pares.

#	Descrição
1	Laço A considerando todos os pontos coletados.
2	Pega próximo ponto do Laço A.
3	Laço B considerando todos os pontos coletados.
4	Pega próximo ponto do Laço B.
5	Filtro que evita que o cálculo da distância de A em relação a B seja reaplicado para B em relação a A. Se A for maior que B, ignora este par e volta para o Passo 4.
6	Utiliza a função de geoprocessamento, obtendo a distância em metros de A em relação a B.
7	Se a distância for maior que 20 metros, então os pontos A e B não estão próximos e o fluxo segue para o próximo ponto do Laço B.
8	Se a distância menor ou igual a 20 metros, então A e B são considerados pontos próximos e são armazenados para compor a lista de pares.
9	Verifica se há mais pontos do Laço B.
10	Verifica se há mais pontos do Laço A.

Tabela 4.6: Detalhamento dos passos do algoritmo 1 (Detecção de Pares).

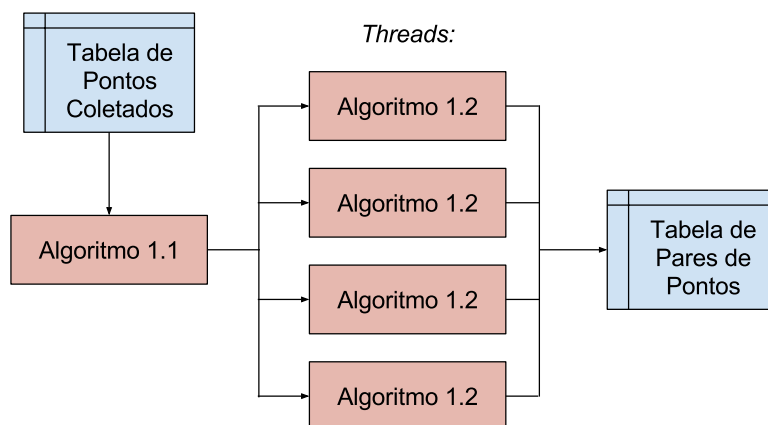


Figura 4.9: Estrutura de execução dos algoritmos 1 e 2.

## 4.2.4 Etapa 2 - Detecção de Grupos

Esta etapa tem como objetivo consolidar os pares de pontos próximos detectados na etapa 1 gerando grupos de pontos. Identifica situações onde os pares de pontos próximos dado pelos dispositivos A e B e pelos dispositivos B e C indicam um grupo formado pelos dispositivos A, B e C.

O Algoritmo de Detecção de Grupos é apresentado na Figura 4.10.

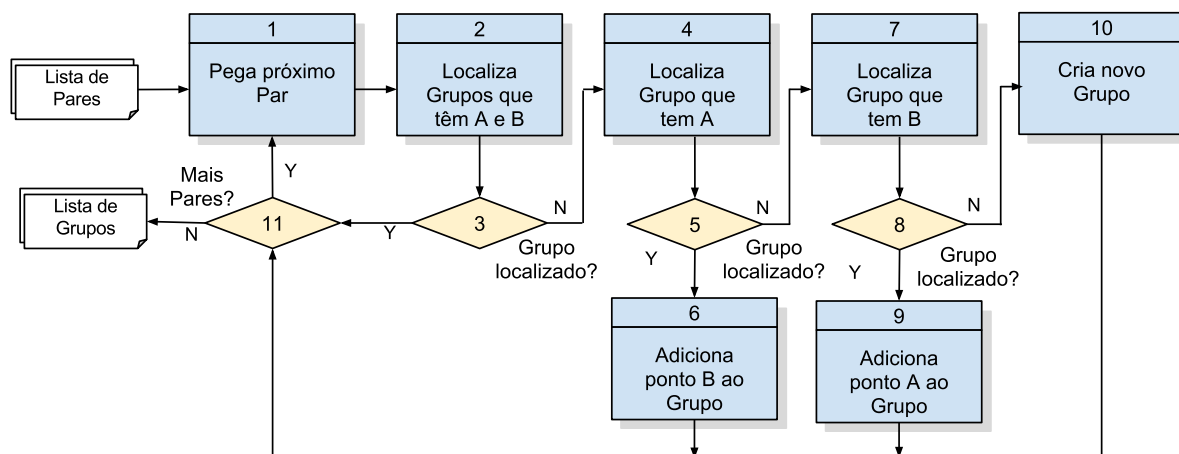


Figura 4.10: Algoritmo 2 - Detecção de Grupos.

O detalhamento do algoritmo 2 encontra-se na Tabela 4.7.

#	Descrição
1	Obtém o próximo para da lista de pares.
2	Verifica se há algum grupo das iterações anteriores que já tenha os pontos A e B.
3	Se for identificado um grupo com os 2 pontos, então nada precisa ser feito e o loop deve prosseguir par próximo par. Este passo elimina eventuais repetições de pares causadas pela expansão da subárea.
4	Verifica se há algum grupo que tenha pelo menos o ponto A.
5	Se o grupo foi localizado, então não possui o ponto B.
6	Adiciona o ponto do dispositivo B ao grupo localizado no item anterior.
7	Verifica se há algum grupo que tenha pelo menos o ponto B.
8	Se o grupo foi localizado, então não possui o ponto A.
9	Adiciona o ponto do dispositivo A ao grupo localizado no item anterior.
10	Se não foi localizado grupo que contenha os dispositivos A ou B, então um grupo novo deve ser criado com o par A e B.
11	Se ainda há par a ser processado, então pegar o próximo par. Se não há, encerra o algoritmo.

Tabela 4.7: Detalhamento dos passos do algoritmo 2 (Detecção de Grupos).

Para que o algoritmo 2 funcione corretamente, os pares devem estar numericamente ordenados, primeiro pela número do dispositivo A, e em seguida pelo número do dispositivo B.

Como resultado da etapa, o algoritmo 2 produz uma coleção de grupos cuja estrutura está descrita na Tabela 4.8.

Atributo	Tipo	Descrição
Grupo ID	Número Inteiro	Número sequencial identificador do grupo
Data/Hora	Data Hora	Data, hora e minuto da coleta dos pontos.
Lista de Dispositivos	Números Inteiros	Lista de dispositivos que compõem o grupo.

Tabela 4.8: Estrutura de saída do algoritmo 2 (Detecção de Grupos).

### 4.2.5 Etapa 3 - Detecção de Riscos

Na etapa 3 são computados dados adicionais acerca dos grupos, sendo eles: i) Duração do encontro, ii) Quantidade média de elementos do agrupamento, iii) Faixa de horário do encontro, e v) Região geográfica do encontro.

O algoritmo deve levar em consideração grupos vigentes identificados no processamento anterior. A Figura 4.11 reflete um exemplo de processamento ocorridos por 5 coletas consecutivas (instantes  $t_0$  a  $t_4$ ), onde o grupo formado pelos dispositivos A e B foi identificado desde o instante  $t_1$  a  $t_3$ . Instantes estes que deverão registrados como horários de início e fim do encontro através do algoritmo 3.

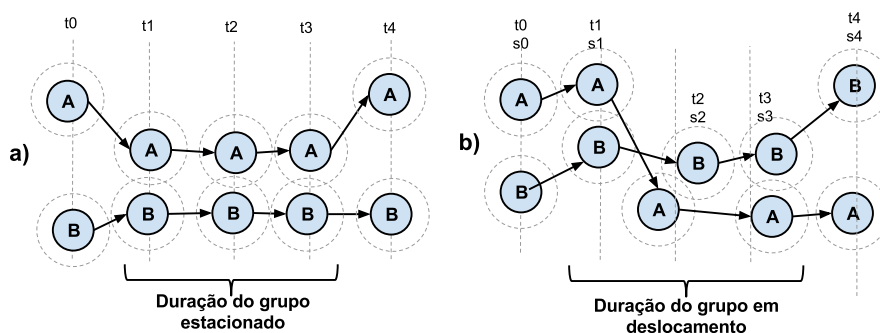


Figura 4.11: Duração do grupo em processamentos consecutivos.

A Figura 4.11.a indica um grupo estacionado (parado) em uma região geográfica, enquanto a Figura 4.11.b representa um grupo em movimento com deslocamentos dos pontos observados a cada processamento ( $s_0$  a  $s_4$ ). O algoritmo 3 deverá atender ambos os casos.

Conforme discutido na Sessão 4.1.2, a precisão dos horários registrados para início e fim do encontro dependem da granularidade da janela de tempo entre uma coleta e outra. Uma janela de tempo reduzida proporciona uma precisão na identificação dos horários de início e fim do agrupamento. Porém, uma janela muito curta compromete o prazo para processamento do lote (Sessão 4.1.3).

Outra definição necessária, é a do tempo mínimo de duração de um encontro para que o mesmo seja considerado uma reunião de pessoas. Na Figura 4.11, estabelecemos este valor em 3 minutos, ou seja, o agrupamento precisa ser detectado por pelo menos 3 processamentos consecutivos para ser considerado um encontro de fato. Encontros com duração menor que a estabelecida, serão descartados em futuras consultas.

Os passos do algoritmo da etapa 3 são mostrados na Figura 4.12.

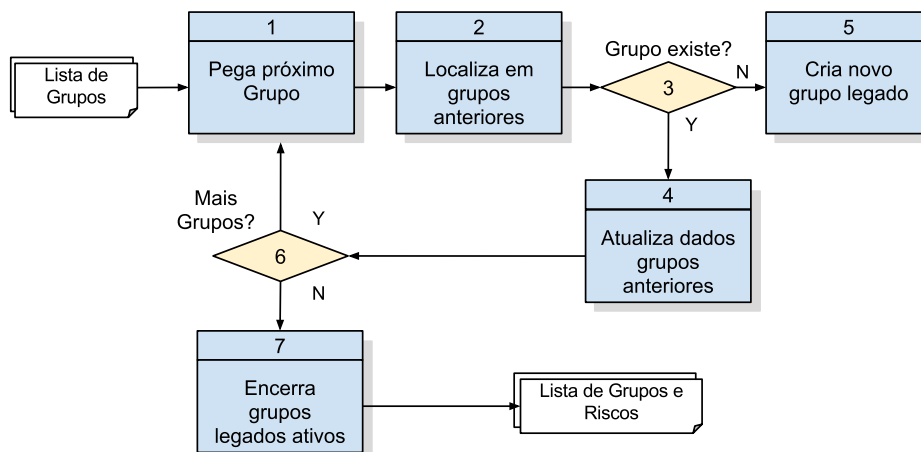


Figura 4.12: Algoritmo 3 - Detecção de Riscos.

Na localização de grupos anteriores do passo 2, a quantidade de elementos comuns entre o grupo atualmente identificado e dos grupos vigentes na base legada definirão se trata-se ou não do mesmo grupo. Os elementos comuns podem ser os mesmos parcial ou totalmente (Sessão 4.1.7).

Para conferir flexibilidade ao algoritmo 3, será considerado que um grupo é o mesmo do legado quando a maioria de seus membros (percentual maior que 50%) são comuns. Esta definição pode ser ajustada na medida que experiências práticas sejam observadas ao longo do tempo.

O detalhamento dos passos do algoritmo 3 está na Tabela 4.9.

Após o processamento de todas as etapas propostas, o resultado produzido é uma coleção de grupos com seus respectivos dados complementares seguindo uma estrutura descrita na Tabela 4.10.

#### 4.2.5.1 Cálculo da duração do encontro

A duração do encontro é computada pela horário inicial registrada na criação de um novo grupo (passo 5) e a horário final registrada quando o mesmo não deixa de ser detectado no processamento atual (passo 7).

Nos caso de grupo vigente, ou seja, que ainda permanece sendo identificado no processamento atual, o horário final permanece em branco (*null*). Desta forma, ao selecionar os

#	Descrição
1	Obtém o próximo grupo do resultado da etapa 2.
2	No conjunto de grupos anteriores vigentes, identificam-se aqueles que possuem pelo menos 50% dos elementos em comum com o grupo identificado.
3	Se não encontrou nenhum grupo compatível, então desvia o fluxo para criação de um novo grupo.
4	Atualiza a quantidade de processamentos pelos quais passou o agrupamento, atualiza a soma de quantidade de elementos e a área geográfica do encontro.
5	Cria um novo grupo considerando o atributo Quantidade de Processamento como 1, o atributo Data/Hora de Início como sendo o momento do processamento atual e o atributo Qtde de Elementos com a quantidade de elementos do grupo.
6	Se houver mais grupos a serem processados, então desvia o fluxo para pegar o próximo grupo. Se não houver, então encerra os encontros vigentes, conforme próximo passo.
7	Grupos vigentes (detectados até o processamento anterior) e que não foram identificados neste processamento devem ser encerrados. Isto é feito atualizando a Data/Hora de finalização do grupo com o horário correspondente ao processamento anterior.

Tabela 4.9: Passos do algoritmo 3 (Detecção de Riscos).

Atributo	Tipo	Descrição
Grupo	Número Inteiro	Identifica unicamente o Grupo.
Data/Hora Início	Data Hora	Data/Hora que o Grupo iniciou a reunião.
Data/Hora Término	Data Hora	Data/Hora que o Grupo se dispersou. Se não preenchido, então o grupo ainda está ativo.
Qtde Processamento	Inteiro	Quantidade processamentos que o Grupo foi detectado. Este dado é necessário para cálculo da quantidade média de dispositivos no grupo.
Qtde Elementos	Inteiro	Soma das quantidade de elementos do Grupo para cálculo da média.
Lista dos dispositivos	Lista de Coordenadas Geográficas	Lista do dispositivos que estiveram dentro do Grupo.
Área Geográfica	Polígono (Retângulo)	Área geográfica comum ao encontro.

Tabela 4.10: Saída do algoritmo 3 (Detecção de Riscos).

grupos na base de grupos legados, deve-se verificar se há ou não a necessidade de incluir os grupos vigentes filtrando os que possuem horário final em branco.

#### 4.2.5.2 Cálculo da quantidade média de elementos

Da a possível variação de quantidade de elementos em um agrupamento, será considerada como fator de risco a quantidade calculada pela média simples entre todas as amostras

processadas. Assim, o passo 4 da Figura 4.12 atualiza o número do processamento (quantidade de vezes que aquele grupo foi detectado em processamentos consecutivos) e soma à quantidade de elementos existentes a quantidade detectada atualmente. O cálculo da média de quantidade de elementos no agrupamento dar-se-á pela divisão da soma de elementos pela quantidade de processamentos.

Sobre os resultados produzidos e armazenados seguindo o leiaute da Tabela 4.10, algumas consultas podem ser executadas extraídos os dados complementares. Estas consultas são descritas a seguir.

#### **4.2.5.3 Frequência dos encontros**

A detecção de um mesmo agrupamento pode ocorrer diversas vezes. A frequência dos encontros é relevante como fator de risco, e por isso é considerado com dado a ser gerado ou obtido na solução do problema.

Diferente da duração do encontro e da quantidade de elementos dos grupos, o cálculo da frequência pode ocorrer a qualquer tempo, tendo em vista que pode ser implementado através uma consulta sobre uma base de dados de agrupamentos (resultado acumulado do algoritmo 3).

#### **4.2.5.4 Horário mais comum dos encontros**

Para implementação do cálculo de faixa de horário mais comum dos encontros de um agrupamento, o dia foi dividido em 24 horas. Para cada um dos encontros, o horário de início foi arredondado para a hora menor e o horário de término para a maior. Exemplo, um encontro que iniciou-se às 10:20 e encerrou-se às 11:25 foi, para fins de processamento, arredondado para o intervalo de 10:00 às 12:00 horas. Este processo facilitará a detecção de horários mais comuns para os encontros.

Em seguida, os horários alterados de todos os encontros são sobrepostos, e o intervalo de horário que apresentar maior quantidade de encontros será eleito como horário comum de encontro (Figura 4.13). Se os encontros apresentarem horários muito dispersos durante o dia, pode ocorrer do grupo não possuir qualquer horário comum.

É relevante informar, além do horário comum de encontro do grupo, qual a sua representatividade em relação aos demais encontros. Logo, um dado adicional deve ser fornecido em percentual, representando a razão entre participação dos encontros daquele horário comparados à quantidade total de encontros.

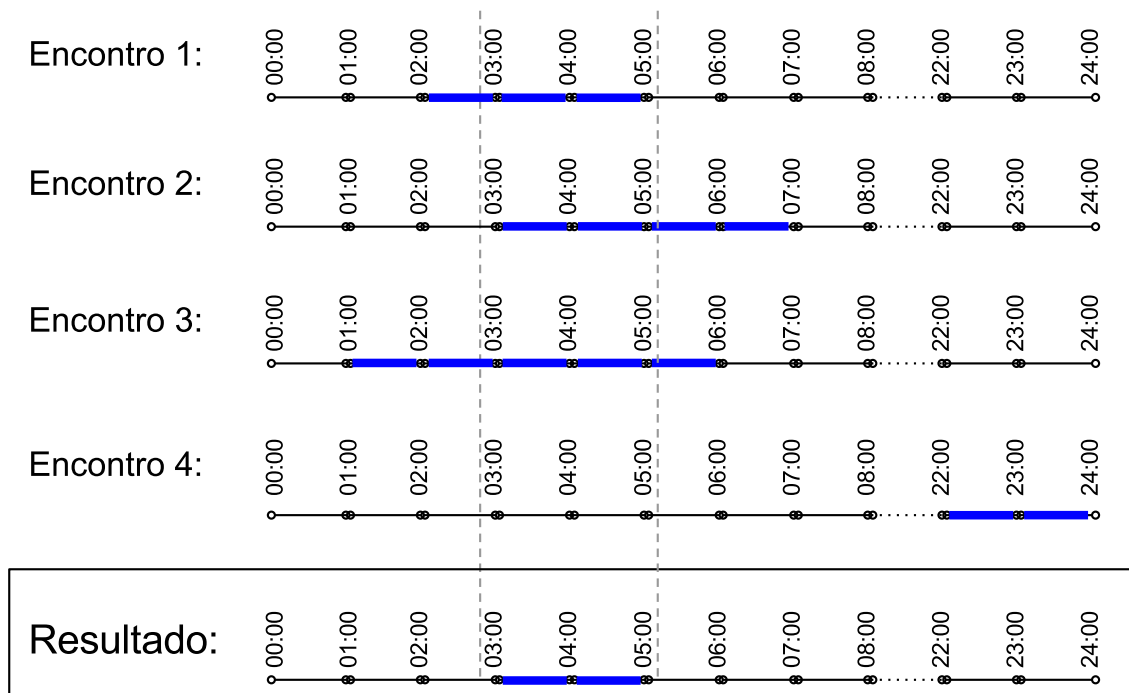


Figura 4.13: Cálculo de horas comuns aos encontros.

#### 4.2.5.5 Área geográfica comum aos encontros

No passo 4 do algoritmo 3, entre os dados atualizado, está a área geográfica comum cobrindo pontos daquele encontro. O registro antecipado da área geográfica do agrupamento facilita consultas que terão apenas que sobrepor as áreas geográficas de vários encontros.

A produção da área geográfica do encontro é dada pelos seguintes passos executados a cada processamento de pontos: i) obtenção de uma área geográfica retangular abrangendo os pontos da amostra; ii) esta área geográfica é sobreposta à anterior produzindo uma nova área através da interseção destas áreas; iii) esta área produzida é atualizada no registro do agrupamento. A Figura 4.14 demonstra este processo.

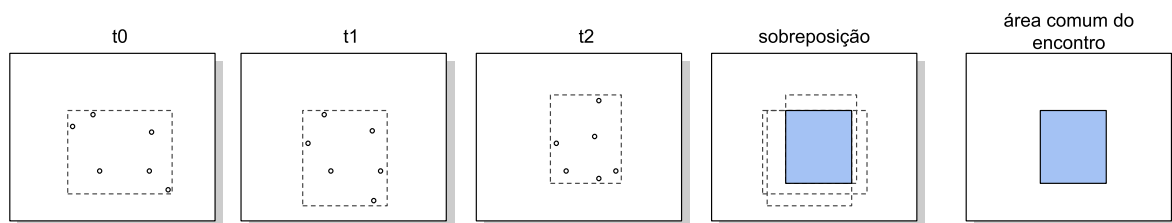


Figura 4.14: Área geográfica comum do encontro.

Se o grupo estiver em deslocamento, a sobreposição da nova área com a anterior causará uma redução no resultado final podendo chegar a eliminar a área comum. O que é conveniente pois realmente não há área comum a considerar neste caso, para agrupamentos em movimento.



Após o estabelecimento da área geográfica comum no âmbito de um encontro (Figura 4.14), os registros ficam disponíveis para várias combinações através de consultas. Entre estas consultas, pode-se obter, por exemplo, uma área geográfica comum aos encontros de um agrupamento. A consulta deve produzir um retângulo que abrange as áreas de todos os encontros. Este retângulo representa a área dos encontros (Figura 4.15).

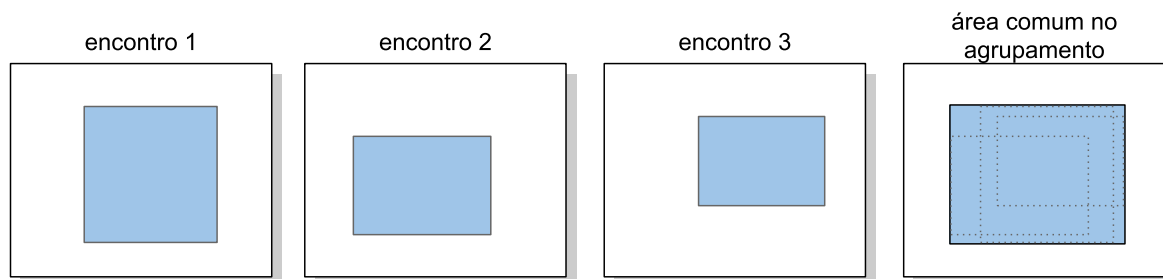


Figura 4.15: Área geográfica comum a todos os encontros.

Definidos os algoritmos e as restrições acerca do problema proposto, o próximo capítulo aborda o resultado da implementação e testes de desempenho.

# Capítulo 5

## Resultados e Análise

Para validar os algoritmos apresentados neste trabalho foram criadas duas bases de dados para testes. A primeira, com número reduzido de registros, tem o objetivo de demonstrar a efetividade dos algoritmos implementados. A segunda, mais volumosa em termos de quantidade de registros, visa aferir o desempenho dos algoritmos. Ambas as bases foram inicialmente simuladas em tabelas através de um sistema gerenciador de banco de dados relacional PostgreSQL na versão 9.4.

Com exceção do algoritmo 1.2 que foi implementado também em linguagem C, todos os algoritmos foram desenvolvidos em linguagem nativa do PostgreSQL PL/pgSQL, levando em consideração que esta última apresenta algumas vantagens como a portabilidade dos programas desenvolvidos, além de ser executado diretamente dentro do servidor de banco de dados com acesso facilitado aos dados através de tabelas. Também permite a chamada a funções geográficas diretamente através da biblioteca PostGIS instalada no PostgreSQL.

Alguns valores iniciais para os parâmetros utilizados nos algoritmos foram previamente estabelecidos. Segue uma lista destas definições.

- a. Período de envio de pontos geográficos (janela de processamento): 1 minuto
- b. Distância mínima entre pontos para definição de proximidade: 20 metros (Conforme abordado na Sessão 4.1.1)
- c. Tempo mínimo de duração do encontro para definição de reunião: 3 minutos (corresponde a pelo menos 3 detecções consecutivas do mesmo grupo)
- d. Quantidade máxima de pontos por subárea: 250, 500, 1000 e 2000
- e. Percentual de elementos em comum para identificar grupos entre processamentos subsequentes  $> 50\%$

Os valores inicialmente definidos estabelecem um cenário experimental. Tais valores deverão receber atualizações futuras de acordo com o retorno de avaliações realizadas na

aplicação prática por equipes externas de investigações ou sistemas de análise de risco. Desta forma, ajustes incrementais devem ser aplicadas sobre estas variáveis até que as mesmas representem um cenário mais efetivo.

O equipamento utilizado para simular o processamento dos algoritmos possui a configuração detalhada na Tabela 5.1.

Recurso	Especificação
Processador	Core i5-2467M 1.6 GHz ( <i>dual core with hyperthread</i> )
Memória	8 Gigabytes
Disco Rígido	Hard Disk 516 GBytes 5400 RPM
Sistema Operacional	Ubuntu 14.04.4
SGBD	PostgreSQL 9.4

Tabela 5.1: Equipamento utilizado nos testes.

Neste capítulo, cada rotina testada foi executada 10 vezes e os valores médios obtidos são os que foram considerados no resultado.

## 5.1 Execução do Algoritmo 1.1 - Divisão do espaço geográfico em subáreas

Para demonstrar a aplicação deste algoritmo foi elaborada uma massa de dados com 20 coordenadas aleatórias mostradas na Figura 5.1. Estes registros formam um cenário onde a área deve ser subdividida em quadrantes através da execução do algoritmo 1.1.

A lista de pontos e suas coordenadas estão detalhados na Tabela 5.2.

Devido à reduzida quantidade e pontos da massa de dados, o parâmetro que define a quantidade máxima de pontos por subárea foi ajustado para 5. Ou seja, no resultado final, cada subárea pode conter até o máximo 5 pontos. Após a aplicação do algoritmo 1.1 sobre este cenário, obteve-se como resultado um total de 6 subáreas, cuja disposição é mostrada na Figura 5.2. No processo de subdivisão, devido à dispersão irregular dos pontos, algumas subáreas podem ficar vazias.

As áreas sobrepostas observadas na Figura 5.2 correspondem à expansão de subáreas necessária para eliminar a interdependência no processamento paralelo (Sessão 4.1.5). Esta sobreposição avança 10 metros sobre as subáreas vizinhas.

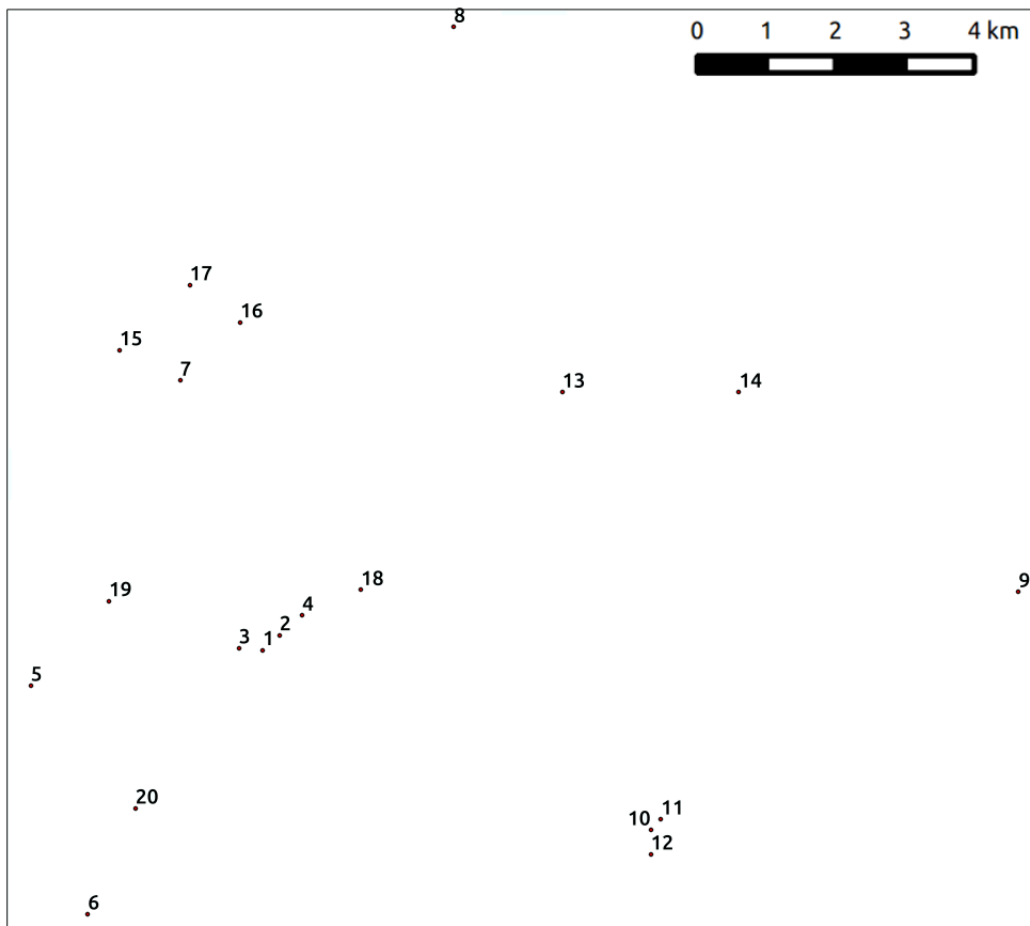


Figura 5.1: Pontos da massa de dados para teste dos algoritmos 1.1 e 1.2.

## 5.2 Execução do Algoritmo 1.2 - Detecção de Pares em Subáreas

Este algoritmo atua sobre cada uma das subáreas geradas pelo algoritmo anterior identificando os pares de pontos através da comparação das distâncias filtrando os pares próximos.

O comando utilizado para executar a função recursiva tem a sintaxe `select unb.pairs_detection(1, 1)`; onde os valores 1 passados como parâmetros indicam que a execução se dará em apenas uma fila de processamento. Como resultado, a tabela `unb.pairs` é populada com os 6 pares de pontos próximos detectados, mostrados na Tabela 5.3. A execução distribuída em várias filas será demonstrada no teste de desempenho pela paralelização.

O par composto pelos dispositivos 1 e 2 consta duplicado no resultado do algoritmo na Tabela 5.3, pois encontram-se próximos entre si e dentro da área de expansão de duas subáreas vizinhas. A execução do algoritmo de agrupamento corrige naturalmente esta anomalia.

Dispositivo	Coordenada (Latitude e Longitude)
1	-51.101676483154 -29.796031666667
2	-51.1015887192601 -29.7959551185365
3	-51.1018001278233 -29.7960222315326
4	-51.1014717009427 -29.7958530542704
5	-51.102878427145 -29.7962165116069
6	-51.1025815151294 -29.7973988785494
7	-51.1021047950715 -29.794637443819
8	-51.1006860298657 -29.7928075237786
9	-51.0977610798922 -29.7957328312539
10	-51.0996673366895 -29.7969634029281
11	-51.0996141489865 -29.7969081173124
12	-51.0996673366895 -29.7970897008321
13	-51.1001253252357 -29.7946979242584
14	-51.0992093481433 -29.7946979242584
15	-51.1024182982438 -29.7944795710083
16	-51.1017933610516 -29.7943357580847
17	-51.102055215758 -29.7941431613534
18	-51.1011681313195 -29.7957212020569
19	-51.102473465208 -29.7957786565766
20	-51.1023350013632 -29.7968519327902

Tabela 5.2: Amostra de dados para teste dos algoritmos 1.1 e 1.2.

Dispositivo A	Dispositivo B
1	2
1	2
1	3
2	4
10	11
10	12

Tabela 5.3: Pares de pontos próximos resultante do algoritmo 1.2.

### 5.3 Execução do Algoritmo 2 - Detecção de Grupos

O algoritmo 2 agrupa os pares de pontos próximos formando conjuntos conforme os dispositivos encontrados na lista dos pares gerados pelo algoritmo 1.2. Após a execução, o conteúdo resultante apresenta os registros mostrados na Tabela 5.4. A coluna relativa a "Lista de dispositivos" é coleção de identificadores de dispositivos.

ID do Grupo	Timestamp	Lista de dispositivos
1	2016-05-24 12:00	1,2,3,4
2	2016-05-24 12:00	10,11,12

Tabela 5.4: Grupos detectados pelo algoritmo 2.

O resultado mostra que os pontos 1, 2, 3 e 4 formam um grupo e os pontos 10, 11 e 12

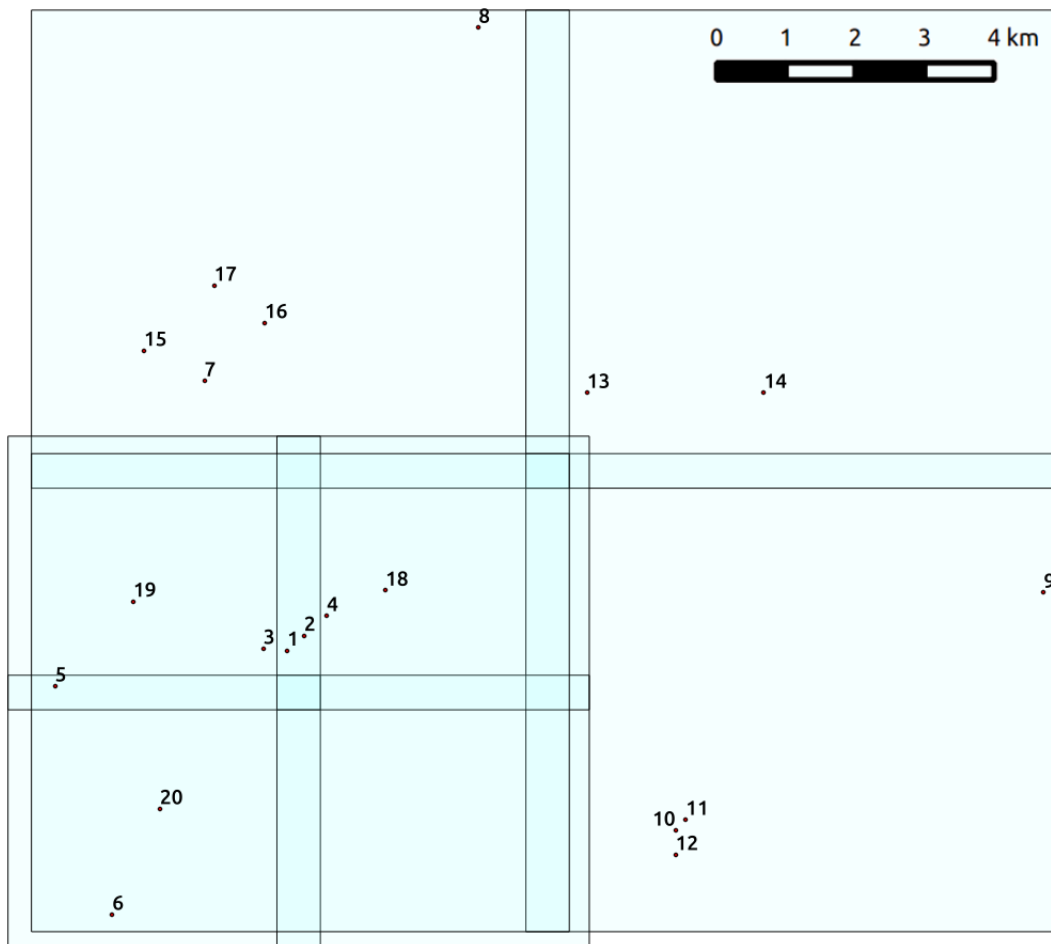


Figura 5.2: Distribuição dos pontos nas subáreas após aplicação do algoritmo 1.1.

formam outro grupo, o que pode ser visualmente observando a Figura 5.2 e verificado na Tabela 5.3.

## 5.4 Execução do Algoritmo 3 - Detecção de Riscos

Para a demonstração do algoritmo de detecção de risco, é necessário acompanhar a evolução dos grupos através de processamentos subsequentes sobre os pontos coletados. Para isso, foi elaborada uma massa de dados complementar que simula a coleta de pontos em instantes consecutivos e processados pela detecção de grupos. A Figura 5.3 mostra diversos instantes simulados (12:00 a 12:05 horas) com a disposição espacial dos pontos em cada momento. As amostras são processadas sequencialmente pelo algoritmo 3.

Na Figura 5.3, durante o período de 12:01 a 12:04, um grupo com dispositivos em comum foi detectado. Mesmo com a aproximação do dispositivo 1 no instante 12:02 e do afastamento do dispositivo 4 no instante 12:04, o grupo permaneceu reunido com a maioria de seus membros (*elementos\_similares* > 50%) durante o período. Tal cenário foi reproduzido através de registros simulados para facilitar a aplicação do código do algoritmo 3.

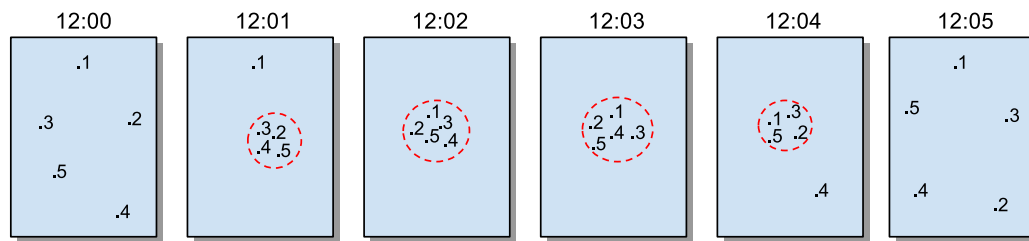


Figura 5.3: Grupos detectados durante período de tempo.

Após a aplicação do algoritmo 3 em nas 6 amostras simuladas, o resultado obtido é mostrado na Tabela 5.5.

ID	Data Hora Início	Data Hora Fim	Qtde Proc.	Tot.Disp.	Lista Disp.
1	2016-05-24 12:01:00	2016-05-24 12:04:00	4	18	1,2,3,4,5

Tabela 5.5: Dados de riscos por grupo resultante do algoritmo 3.

A duração do encontro é registrado pelo conteúdo das colunas "Data Hora Início" e "Data Hora Fim". O conteúdo das colunas "Qtde Proc" e "Tot.Disp." são cumulativos e referem-se a quantidade de processamentos executados e ao total de dispositivos acumulados, respectivamente. Com estas colunas, pode-se calcular a quantidade média de elementos do agrupamento conforme a fórmula da Eq. 5.1, que no cenário proposto resultou em 4,5 elementos.

$$qtde\ media\ de\ elementos = \frac{total\ de\ dispositivos}{qtde\ de\ processamentos} \quad (5.1)$$

Para algumas situações, em vez da construção de um algoritmo para computar ou transformar os dados através de um processamento, verificou-se que a aplicação de linguagem de alto nível sobre um banco de dados (*Structured Query Language – SQL*) atende a necessidade de geração dos dados requeridos em tempo de execução inferior a 1 segundo. É o caso dos cálculos de frequência dos encontros, dos horários e das áreas geográficas comuns.

Assim, foram elaboradas consultas para cada um dos casos, conforme abordados nas próximas sessões. Nada impede, no entanto, que em futuras evoluções da solução tais consultas sejam convertidas para outras linguagens de programação.

### 5.4.1 Cálculo da frequência dos encontros

Para ilustrar o cálculo da frequência dos encontros, foi elaborada uma consulta que lê uma tabela com todos os agrupamentos detectados e lista a quantidade de vezes que um determinado grupo se encontrou. Para demonstração, populamos uma tabela `unb.group_risk` com o conteúdo mostrado na Tabela 5.6.

Ao aplicar a consulta sobre os dados da tabela, passando como parâmetro o ID do grupo igual a 1, é retornado o valor 2, que corresponde aos 2 encontros realizados pelo mesmo

ID	Data Hora Início	Data Hora Fim	Qtde Proc.	Tot.Disp.	Lista Disp.
1	2016-05-24 12:35:00	2016-05-24 16:24:00	229	916	1,2,3,4,5
2	2016-05-25 12:01:00	2016-05-25 12:31:00	30	90	10,11,12
3	2016-05-25 14:10:00	2016-05-25 20:21:00	371	1855	1,2,3,4,5,6
4	2016-05-26 15:45:00	2016-05-26 15:47:00	2	9	1,2,3,4,5
5	2016-05-27 12:01:00	2016-05-27 12:04:00	3	9	20,21,22
6	2016-05-28 21:07:00	2016-05-28 22:55:00	108	486	1,2,18

Tabela 5.6: Dados de entrada para consulta de frequência de encontros.

grupo identificados pelos IDs 1 e 3. O ID 4, apesar de representar o mesmo grupo pela similaridade de seus membros, possui apenas 2 minutos de duração e portanto foi descartado da consulta uma vez que o limite mínimo é de 3 minutos para ser considerado um encontro. O ID 6 também foi descartado pois apesar de alguns elementos em comuns, os mesmos não correspondem a mais de 50% de similaridade.

#### 5.4.2 Cálculo do horário mais comum dos encontros

A consulta elaborada em linguagem SQL compara os horários de todos os encontros de um determinado agrupamento listando o intervalo mais frequente em horas, de acordo com a política discutida em 4.2.5.4.

A mesma tabela populada para o teste da consulta de frequência (Tabela 5.6) pode ser usada para o teste da consulta de extração dos horários comuns. Desta forma, aplicou-se a consulta utilizando como referência o grupo relativo ao ID 1. Assim como o levantamento de frequência, esta consulta também filtra grupos cuja duração seja menor que 3 minutos e que os elementos em comum seja inferior a 50%.

O resultado da consulta é uma matriz com os valores 14, 15, 16 e 17 correspondentes aos horários que os encontros de ID 1 e 3 têm em comum (14:00 às 17:00 horas).

#### 5.4.3 Cálculo da área geográfica comum dos encontros

Esta consulta em SQL baseia-se nos mesmos filtros da consulta de cálculo de frequência dos encontros, gerando uma área retangular envolvendo todas as áreas comuns previamente calculadas no passo 4 do algoritmo 3. A execução da consulta retornou um registro com uma área geográfica retangular abrangendo às áreas comuns dos encontros correspondentes aos ID 1 e 3.



## 5.5 Execução dos testes de desempenho

A base para testes de desempenho foi produzida por simulação através da criação de variantes de alguns trajetos obtidos por equipamentos GPS. Novos trajetos foram adicionados através do deslocamento aleatório horizontal e vertical dos pontos originais no espaço geográfico, aumentando a quantidade de pontos da amostra. Deste resultado, ainda foram criados novos trajetos pela inversão das latitudes e longitudes simulando novos pontos. Foram separados três conjuntos de 10.000 pontos que correspondem a três minutos consecutivos simulando a coleta de pontos das tornozeliras nos horários 12:00, 12:01 e 12:02.

A quantidade de pontos inicialmente estabelecido em 10.000 foi estimada observando a quantidade de indivíduos monitorados em 2015 em um dos estados brasileiros que era de aproximadamente 1.200. Considerando a adesão dos demais estados em um sistema de monitoramento eletrônico de abrangência nacional, o referido valor foi adotado como patamar mínimo. Este valor poderá crescer tendo em vista a população carcerária maior que 500 mil indivíduos (1.1) aumentando a relevância de se manter o desempenho dentro da janela temporal disponível.

Os tempos de execução que compõem os gráficos neste capítulo foram obtidos através da submissão dos *scrits* através de um comando no nível de sistema operacional denominado *time*. Este comando conta o tempo decorrido entre a submissão do comando e seu término.

### 5.5.1 Cenário 1 – Processamento em PL/pgSQL sem paralelismo

Este cenário foi concebido para demonstrar o funcionamento de todos os algoritmos sem usar o recurso de paralelismo. Foram executados os algoritmos 1.2, 2 e 3. O algoritmo 1.1 que subdivide a área em subáreas não foi empregado uma vez que não haverá execução paralela do algoritmo 1.2. Este foi executado apenas uma vez sobre a área inteira contendo os 10.000 pontos.

#### 5.5.1.1 Algoritmo 1 - Detecção de Pares

Executando o algoritmo 1.2 contra uma das amostras da base de dados com uma amostra 10.000 pontos relativa a 12:00 horas, obteve-se como resultado uma quantidade aproximada de 9100 registros, levando em média 280 segundos para ser concluído.

#### 5.5.1.2 Algoritmo 2 - Detecção de Grupos

Para detecção de grupos de pontos próximos, o algoritmo 2 gerou uma tabela com 2.414 grupos detectados em aproximadamente 2,6 segundos. O tempo de execução mostrou-se reduzido, mesmo para a quantidade de 9100 de pares de pontos detectados na sessão anterior.

### 5.5.1.3 Algoritmo 3 - Detecção do Riscos

Os algoritmos 1.2 e 2 foram então aplicados às 3 amostras de 10.000 pontos cada gerando um conjunto de grupos para cada processamento. Intercalado a estas execuções foi executado o algoritmo 3 apresentando como tempo médio de execução de 2,8 segundos

### 5.5.1.4 Tempo de Execução por Algoritmo

O algoritmo que detecta os pares de pontos próximos apresentou tempo de processamento muito superior às demais etapas relativas aos algoritmos 2 e 3, alcançando o tempo médio de 280 segundos (Figura 5.4). Ao considerar o intervalo de tempo de janela de processamento inicialmente estipulado um minuto, a soma do processamento das três etapas seria muito superior tornando este cenário inviável.

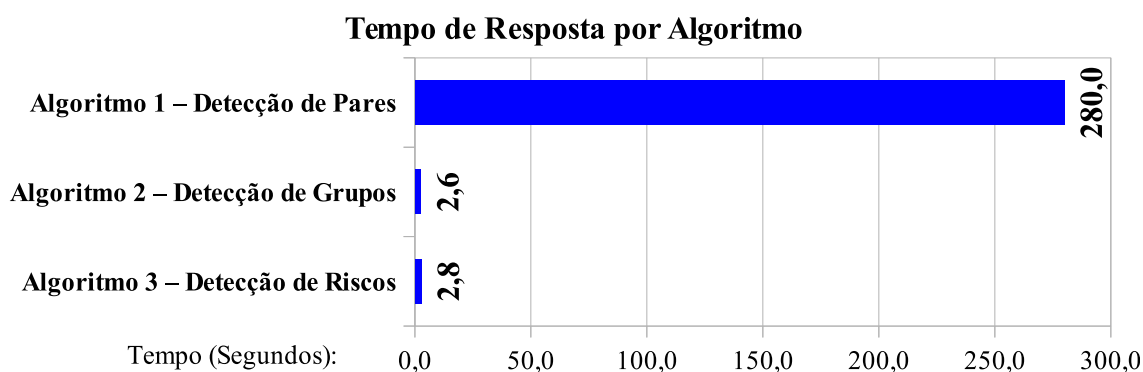


Figura 5.4: Tempo de execução por algoritmo sem paralelismo.

Para evitar que a quantidade de monitorados, e por consequência a quantidade de pontos coletados, comprometa o prazo da janela de tempo disponível para processamento, suger-se algumas alternativas possíveis: i) a atualização (*upgrade*) do equipamento (*hardware*) responsável pelo processamento (escalonamento vertical) até que o tempo de processamento fique abaixo do máximo estipulado; ii) execução paralelizada do algoritmo 1.2 através da subdivisão da área pelo algoritmo 1.1; e iii) desenvolver o algoritmo 1.2 em linguagem de melhor desempenho que o PL/pgSQL.

A adoção do atualização do equipamento como solução é a mais simples pois não apresentaria impacto significativo na complexidade da arquitetura da solução. Porém, embora a evolução dos computadores tenha alcançado grandes avanços em termos de velocidade de processamento, tal direção limita o crescimento do sistema.

As alternativas ii) e iii) serão abordadas nos próximos cenários.

## 5.5.2 Cenário 2 - Processamento em PL/pgSQL com paralelismo

Neste cenário, motivado pelo baixo desempenho no tempo de execução da etapa de detecção de pares, foi aplicado o algoritmo de divisão em subáreas (algoritmo 1.1) e em seguida a execução paralela do algoritmo 1.2 (Figura 4.9).

Ao distribuir os pontos em subáreas, fez-se necessário definir a quantidade máxima de pontos por subárea. Tendo por base uma tabela de 10.000 pontos correspondente à primeira amostra, foram experimentados números máximos de pontos por subárea com os valores 250, 500, 1000 e 2000 para verificação do impacto no desempenho.

### 5.5.2.1 Algoritmo 1.1 - Distribuição de Pontos em Subáreas

Este algoritmo foi elaborado em PL/pgSQL conforme os passos mostrados em Figura 4.6 e Figura 4.7. Os tempos de execução variaram entre 1,4 e 4,6 segundos (Figura 5.5).

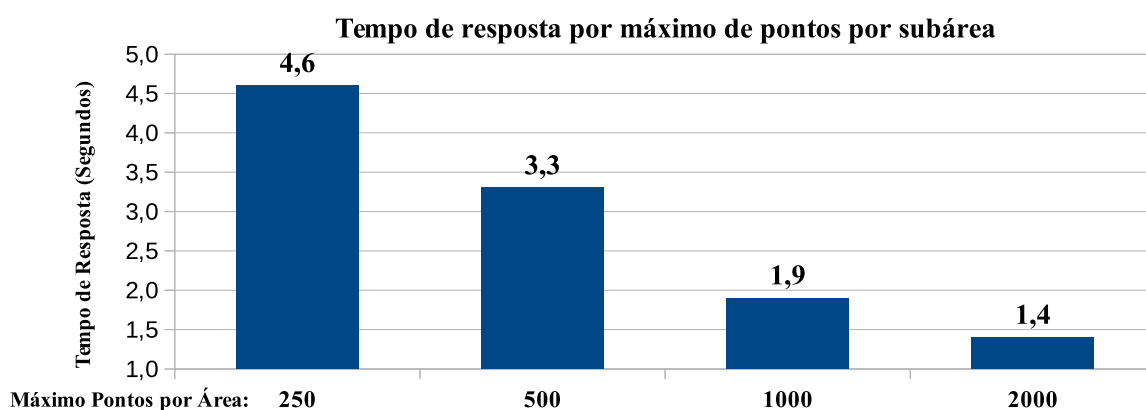


Figura 5.5: Tempo de execução por máximo de pontos por subárea.

Quanto maior a quantidade máxima de pontos por área, menor o número de áreas geradas e por consequência, menor o tempo de execução do algoritmo 1.2. No menor dos casos, a área tem aproximadamente 150 metros quadrados (Tabela 5.7).

	Máximo de Pontos por Área			
	2000	1000	500	250
Quantidade de Áreas	22	37	91	151
Tamanho Médio das Áreas (m <sup>2</sup> )	6.662.905,40	3.961.789,13	1.610.986,67	970.874,14
Menor Tamanho de Área (m <sup>2</sup> )	9.023,61	2.275,65	578,85	149,74
Maior Tamanho de Área (m <sup>2</sup> )	36.643.506,10	36.643.506,10	36.643.506,10	36.643.506,10

Tabela 5.7: Máximo de pontos por subárea e tamanho das subáreas.

A área total tratada corresponde a 146,56 km<sup>2</sup>. O tamanho da maior subárea é constante em todos os casos devido ao quadrante da primeira subdivisão que possui uma quantidade de pontos inferior a 250.

### 5.5.2.2 Algoritmo 1.2 - Detecção de Pares em Paralelo

Como o PostgreSQL não fornece suporte direto ao desenvolvimento de rotinas para execuções paralelas (threads) em código PL/pgSQL, utilizou-se o apoio de um *script shell* executado no sistema operacional do ambiente de teste para o escalonamento das tarefas.

Aplicou-se um método de escalonamento Round-Robin para a distribuição das subáreas sobre as filas. Neste método, as subáreas são sequencialmente atribuídas a cada fila disponível. As subáreas que sobraram são novamente distribuídas sequencialmente. Este processo se repete até que todas as subáreas tenham sido atribuídas às filas.

Esta solução segue o tipo de paralelismo Simples Instrução e Múltiplos Dados (*Single Instruction, Multiple Data – SIMD*) conforme abordado na sessão 2.3.1.2

### 5.5.2.3 Algoritmos 3 e 4 - Detecção de Grupos e Detecção de Riscos

Diante do baixo tempo de execução apresentado no cenário 1, menor que 3 segundos cada, os algoritmos 3 e 4 não serão reexecutados no cenário 2 assumindo os valores medidos naquele cenário. O mesmo se aplica aos cenários 3 e 4.

### 5.5.2.4 Tempos de Execução por Máximos Pontos por Área

Comparando o tempo de execução paralela utilizando o equipamento usado nos testes, levando em consideração os quatro valores experimentados para definição de máximo de pontos por área (250, 500, 1000 e 2000) e quantidades de filas de processamento de 2, 4, 8 e 16, segue o resultado obtido na Figura 5.6.

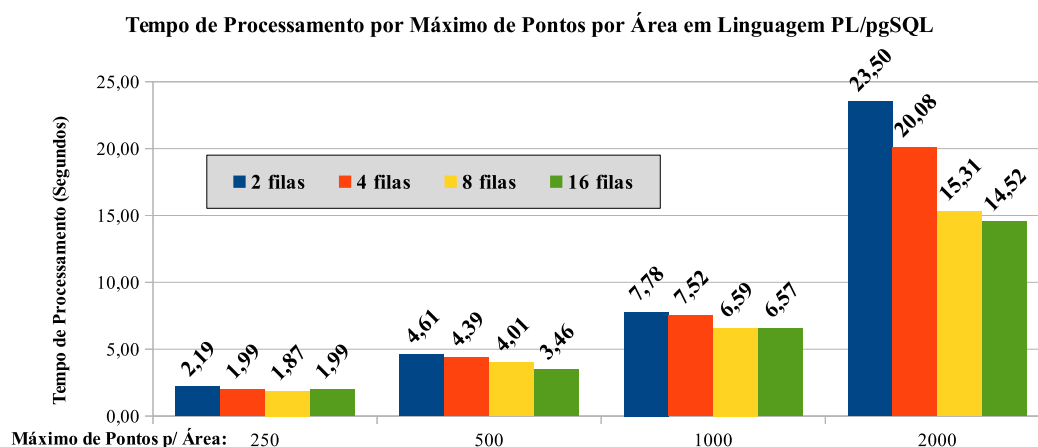


Figura 5.6: Tempo de execução do algoritmo de detecção de pares em linguagem PL/pgSQL.

Pelo desempenho apresentado na Figura 5.6, a medida em que se diminui a quantidade de pontos máximos por área, também reduz o tempo de processamento devido a menor quantidade de comparações entre pontos e cálculos da distancia. Para a amostra utilizada, a redução do tempo foi significativa, tendendo a estabilizar por volta de 250 pontos por área.

Ao escolher o melhor valor para máximo de pontos por área, deve-se levar em consideração o tempo de processamento gasto na subdivisão da área e o tempo de processamento de detecção de pares de pontos próximos, pois na medida que o tempo de um diminui, o outro aumenta.

Para as subáreas com maior número de pontos (1000 e 2000), a quantidade de 8 ou 16 filas apresentou melhor resultado. A diferença, no entanto, permaneceu pequena para áreas definidas com menor quantidade de pontos.

Ao analisar o gráfico, deve-se levar em consideração que no banco de dados há o custo inicial de preparação do código antes de sua execução, apresentando melhores resultados para situações com menor número de subáreas.

### 5.5.3 Cenário 3 - Algoritmo 1.2 em Linguagem C sem paralelismo

Neste cenário, o algoritmo 1.2 de detecção de pares foi em linguagem C. Neste caso, não se aplicou a execução paralelizada da rotina, ou seja, o processo foi realizado de forma serial. A rotina leu os 10.000 pontos de um arquivo texto em sistema de arquivos (*file system*) e grava o resultado também como arquivo texto. Os dados que deram origem ao arquivo de entrada foram extraídos da tabela correspondente no banco de dados.

O tempo de execução do processamento da rotina em linguagem C foi de 14.66 segundos (Figura 5.7), cerca de 20 vezes mais rápido que a mesma rotina em PL/pgSQL, que foi de 280 segundos.

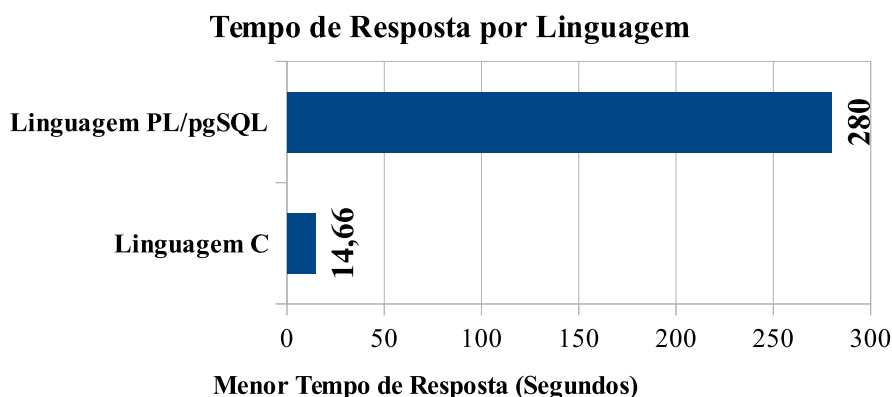


Figura 5.7: Tempo de execução por linguagem.

### 5.5.4 Cenário 4 - Algoritmo 1.2 em Linguagem C com paralelismo

Utilizando a rotina em Linguagem C elaborada no cenário anterior, separando o processamento por subárea e experimentando a execução com 2, 4, 8 e 16 filas de processamento, usando os mesmos parâmetros do cenário 2, obteve-se o gráfico mostrado na Figura 5.8.

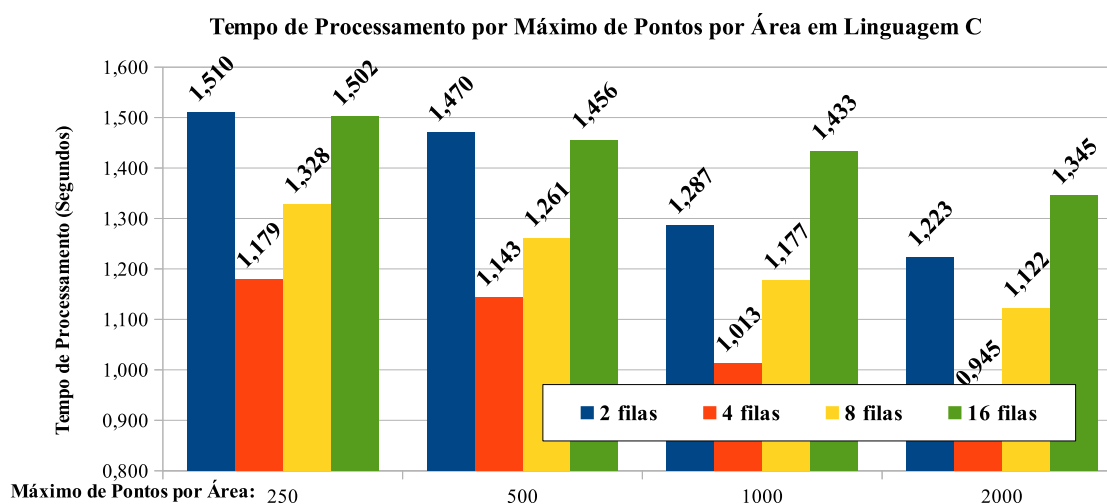


Figura 5.8: Tempo de execução do algoritmo de detecção de pares em linguagem C.

Os tempos obtidos para a execução paralela da rotina em linguagem C são, no geral, bem inferiores aos obtidos em linguagem PL/pgSQL, permanecendo abaixo de 2 segundos. As diferenças são significativas em relação à quantidade de filas de processamento, apresentando melhor desempenho quando executado em quatro filas. O tempo de execução inferior obtido com quatro filas justifica-se devido a arquitetura do processador do equipamento utilizado no teste, um *Core i5-2467M* (Tabela 5.1). Este equipamento possui quatro núcleos lógicos com a tecnologia *Hyperthread* (dois núcleos físicos). Na rotina em Linguagem C, o código é previamente compilado e o custo inicial para iniciar o processo é insignificante diante do tempo total do processamento, evidenciando o melhor desempenho com 4 filas de processamento.

Quando se compara o processamento paralelo do algoritmo 1.2 em linguagem PL/pgSQL, os gráficos das Figura 5.6 e 5.8 apresentam um comportamento diferente que vai além dos valores máximos e mínimos obtidos. O algoritmo em linguagem PL/pgSQL mostrou-se mais sensível à quantidade de pontos por subárea seguida da quantidade de filas e processamento. Esta diferença é provocada pela maior dificuldade da linguagem PL/pgSQL em lidar com a combinação ponto a ponto através dos laços (*joins*) antes da etapa do cálculo da distância e filtrar. Pelo mesmo motivo, independente da quantidade de processadores lógicos disponíveis, dividir as subáreas em mais filas apresentou um melhor desempenho médio.

O mesmo algoritmo em linguagem C é mais sensível à quantidade de filas de processamento seguido da quantidade de pontos por subárea. A linguagem C trabalha mais eficientemente reduzindo o tempo de resposta na etapa de combinação dos pontos ressaltando a melhoria do desempenho quando submetido a 4 filas de processamento, uma vez que esta configuração aproveita melhor os 4 núcleos lógicos do equipamento. Com a redução do tempo gasto na combinação dos pontos, o algoritmo é mais afetado pela quantidade de subáreas a processar, revelando uma leve redução no tempo de processamento quando o número de pontos máximos por área se eleva.

### 5.5.5 Cenário 5 - Processamento distribuído em *Big Data* com *Hadoop*

O processamento paralelo apresentou melhor desempenho com relação ao processamento serial, tanto em linguagem PL/pgSQL quanto em linguagem C. Desta forma, é natural pensar em uma experimentação em ambiente de processamento distribuído numa arquitetura orientada a *Big Data*. Neste contexto, uma das plataformas mais amplamente utilizadas é a que utiliza *Hadoop* como base. Porém, convém considerar algumas características deste tipo de processamento:

a) *Big Data* presume uma quantidade massiva de dados a serem processados. Este não é o caso descrito neste trabalho. Apesar de implicar em forte processamento, a quantidade de dados processados por vez de 10.000 pontos, não é uma quantidade significativa em termos de volume de dados. Como resultado, sem nenhuma configuração especial, ao carregar os dados em um ambiente de múltiplos nós previstos na plataforma, os dados carregados neste volume tende a se concentrar em um único nó, eliminando a possibilidade de processamento distribuído. Como abordado por [Davenport 2014], o termo *Big Data* é fortemente apoiado em Volume, Variedade e Velocidade. As características de Volume e Variedade não são atendidas em nosso problema.

b) Na prática, segundo alguns testes executados, o processo todo leva alguns minutos para se preparar para o processamento. Este tempo inicial inviabiliza o atendimento da demanda inicial descrita em nosso problema pois extrapola a janela temporal de 1 minuto.

## 5.6 Análise do Resultado

Verificou-se inicialmente o funcionamento dos algoritmos aplicados sobre uma base controlada de poucos registros para evidenciamento da efetividade da lógica proposta durante a elaboração dos algoritmos. Os cenários e as massas de dados foram propostos de forma a tornar claro o funcionamento dos algoritmos. O desenvolvimento dos algoritmos em PL/pgSQL apresentou resultados esperados com baixo nível de programação, o que não impede, no entanto, a codificação dos algoritmos em linguagens como Python, Java, etc.

Com relação aos testes de desempenho utilizando os três conjuntos de 10.000 pontos, foram aplicados os algoritmos apresentando os respectivos tempos de execução nos diversos cenários. No primeiro cenário, abordamos a aplicação dos algoritmos utilizando a linguagem PL/pgSQL apoiado pela biblioteca PostGIS. No segundo cenário, em função do baixo desempenho do primeiro algoritmo, este foi dividido em duas etapas buscando para execução paralela. Neste cenário, buscou-se uma solução onde a distribuição de pontos em subáreas da área original permitindo a paralelização do processamento. O paralelismo apresentou resultado aceitável com relação ao desempenho.

No terceiro cenário, executamos novamente a consulta sem paralelismo, mas desta vez implementando o algoritmo de detecção de pares em linguagem C. Verificamos que o tempo

apresentou uma redução de tempo aceitável. No quarto cenário, abordamos o processamento do algoritmo de detecção de pares em linguagem C de forma paralela. Novamente, a utilização da Linguagem C apresentou tempos de execução ainda menores em função do paralelismo aplicado.

No quinto cenário, tratamos do assunto sobre aplicabilidade dos algoritmos apresentados através de solução de Big Data em plataforma como *Hadoop*. Embora o processamento seja relativamente alto, a quantidade de dados tratados é baixa tornando o Big Data inadequado como alternativo de solução para o problema.

Ao compararmos os menores tempos de processamento em cada cenário, considerando o tempo de execução das 3 etapas, chegamos ao gráfico da Figura 5.9. O Cenário 1 mostrou-se inviável pelo tempo de reposta ultrapassando 285 segundos. Os demais cenários atendem ao requisito de execução abaixo dos 60 segundos colocados como limite. O melhor tempo foi obtido no cenário 4 (7.75 segundos) que combina a execução paralela com o algoritmo de detecção de pares em linguagem C, tornando este cenário o mais recomendado para aplicação.

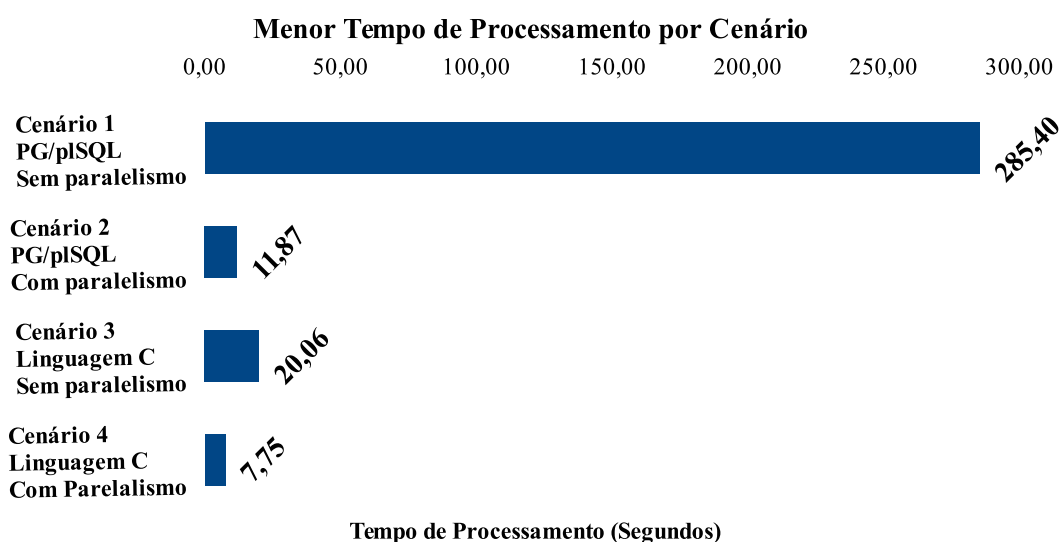


Figura 5.9: Menor tempo de processamento por cenário.

Vale observar que o cenário 3, embora tenha alcançado um tempo de execução aceitável, possui restrições quanto ao crescimento de número de dispositivos monitorados ou redução do tempo de execução. Isto deve-se ao fato que os novos processadores nos últimos anos tem aumentado o número de núcleos e restringido o aumento de frequência. Logo, as alternativas de paralelismo apresentadas nos cenários 2 e 4, tendem a ser mais recomendadas.

Não foi abordada neste trabalho a integração das rotinas em Linguagem C chamadas através de funções na linguagem PL/pgSQL, característica suportada pelo PostgreSQL 9. Estima-se, porém, que os tempos de execução das rotinas nesta situação fiquem próximos às medidas no respectivo cenário.



# Capítulo 6

## Conclusão

O processo de monitoramento de sentenciados por tornozeleira eletrônica pode ser melhorado através da adição de dados que suportem análise de risco para tomada de decisões. O desafio de detectar a reunião de grupos de monitorados acompanhando o tempo de permanência do grupo e a quantidade de elementos foi proposta com adição de restrição de tempo de processamento prefixado. O processamento utilizando algoritmos seriais mostrou-se um problema face ao alto tempo de execução. O maior tempo de processamento concentrou-se na etapa de separação de pares de pontos próximos.

A solução proposta para aumentar o desempenho foi a divisão do espaço geográfico total, com todos os pontos, em áreas menores de forma que cada subárea pôde ser processada em núcleos diferentes permitindo o paralelismo na detecção de pontos próximos. Tal medida implicou em tratar a situação em que pontos próximos estivessem em subáreas adjacentes não sendo detectados, o que foi resolvido através da expansão de cada subárea em 10 metros em todas as direções e a eliminação natural da duplicação no agrupamento dos pontos próximos.

A adoção imediata de rotinas utilizando a linguagem PL/pgSQL para implantação dos algoritmos, por si só atenderia o prazo previsto de 1 minuto. Porém, utilizando linguagem de baixo nível como a Linguagem C, verifica-se que o tempo de execução no geral sofre uma substancial redução (Figura 5.7).

Tal redução no tempo de execução, no entanto, não justificaria abrir mão do paralelismo no processamento proposto, mesmo que o tempo de execução da rotina em Linguagem C, sem o uso do paralelismo, tenha ficado 14.66 segundos correspondendo a aproximadamente 1/4 da janela definida de um minuto. Um eventual aumento na quantidade de pontos a processar (Figura 4.2) ou redução da janela temporal pode comprometer o prazo disponível para processamento. Assim, mesmo com a redução do tempo de execução pela Linguagem C, convém adotar a solução com processamento paralelo.

Diferente de outros algoritmos, a solução proposta neste trabalho contempla o acompanhamento do grupo no decorrer do tempo, atualizando os dados de cada grupo subsidiando a

análise de riscos baseada na duração do encontro (tempo que o grupo permaneceu reunido), na quantidade média de elementos do agrupamento, quantidade de vezes que o grupo se encontrou, horários e regiões geográficas comuns nos encontros do agrupamento.

Os algoritmos propostos se mostraram efetivos na captura do fenômeno de formação de grupos de pessoas portadoras de tornozeleiras eletrônicas. A necessidade de cuidados na implementação foi determinante na obtenção de desempenho apropriado, em particular à monitoração voltada à aplicação da Lei.

## Trabalhos Futuros

Nos testes efetuados sobre a aplicação dos algoritmos desenvolvidos assumiu-se valores inicialmente definidos para vários parâmetros. Na evolução do estudo, vale experimentar outros valores analisando seu impacto tanto em termos de desempenho quanto de precisão dos resultados. Além disso, vale pesquisar e propor mecanismos de auto-detecção da precisão das tornozeleiras para ajustar dinamicamente o parâmetro relativo à distância mínima na definição de proximidade entre elementos.

O escalonamento adotado distribui o processamento das subáreas de acordo com a quantidade de filas de processamento disponíveis utilizando o método Round-Robin. A quantidade de filas representa a quantidade de nós de processamento ou de processadores disponíveis no sistema. Tendo em vista a alta irregularidade na distribuição dos pontos nas subáreas, o método Round-Robin não é o mais eficiente no balanceamento da carga entre os nós. Como trabalho futuro, a adoção de uma solução que distribui dinamicamente as subáreas conforme a disponibilidade dos nós aproveitará melhor os recursos de processamento.

Os códigos foram desenvolvidos utilizando, na sua maioria, a linguagem PL/pgSQL pela facilidade na manipulação dos dados em tabelas e devido ao resultado com reduzido esforço de programação. É possível, no entanto, que os algoritmos sejam futuramente codificados utilizando linguagens mais flexíveis como PHP, Python e Java entre outras. Uma outra alternativa é o desenvolvimento de todos os algoritmos em linguagem C visando reduzir ainda mais o tempo de execução.

Alguns trabalhos a respeito de geoprocessamento e combate à criminalidade tem focado em sistemas de estatística de áreas mais perigosas baseados em contribuições de voluntários que registram ocorrências através de celulares, *tablets* ou computadores. Os mapas com a identificação de áreas e as respectivas ações criminosas ocorridas são geralmente disponibilizados em sites vinculados aos próprios sistemas. Integrar os dados destas ocorrências criminosas com dados sobre agrupamentos de monitorados que reuniram-se nos mesmos locais e horários, pode resultar em dados mais detalhados para equipes de investigação. Desta forma, tal integração entre sistemas pode representar um importante tema a ser desenvolvido em trabalhos futuros.

Também é importante estudar a viabilidade na aplicação da solução desenvolvida neste trabalho no estudo de comportamento de agrupamentos de animais monitorados em áreas de preservação (pesquisas biológicas). Alguns animais selvagens, principalmente os que estão em risco de extinção, podem ter seu deslocamento monitorado por dispositivos com GPS. A detecção de agrupamentos de animais monitorados, quantidade de elementos do grupo e frequência de encontros, etc. pode colaborar em pesquisas sobre o comportamento desses animais [Handcock et al. 2009].

# Referências Bibliográficas

- [Akyildiz et al. 2002] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer networks*.
- [Amdahl 1967] Amdahl, G. M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference*. ACM.
- [Barbosa 2010] Barbosa, R. M. (2010). O monitoramento eletrônico para presos de baixa periculosidade. Monografia, Universidade Católica de Brasília.
- [Câmara and Monteiro 2004] Câmara, G. and Monteiro, A. M. V. (2004). Conceitos básicos em ciência da geoinformação. *Análise espacial de dados geográficos*.
- [Carlino et al. 2013] Carlino, G. A., Carr, J. K., et al. (2013). Clusters of knowledge: R&D proximity and the spillover effect. *Business Review - Federal Reserve Bank of Philadelphia*.
- [Castanedo 2013] Castanedo, F. (2013). A review of data fusion techniques. *The Scientific World Journal*.
- [Chew 1989] Chew, L. P. (1989). Constrained delaunay triangulations. In Springer, editor, *Algorithmica*.
- [Dangermond and Morehouse 1987] Dangermond, J. and Morehouse, S. (1987). Trends in hardware for geographic information systems. In *Proceedings of the Eighth International Symposium on Computer-Assisted Cartography (Auto-Carto 8), Baltimore, Maryland, March*.
- [Daubal et al. 2013] Daubal, M., Fajinmi, O., Jangaard, L., Simonson, N., Yasutake, B., Newell, J., and Ali, M. (2013). Safe step: a real-time GPS tracking and analysis system for criminal activities using ankle bracelets. *21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*.
- [Davenport 2014] Davenport, T. (2014). *Big Data no Trabalho*. Harvard Business Review Press.

- [Departamento Penitenciário Nacional - DEPEN 2014] Departamento Penitenciário Nacional - DEPEN (2014). Levantamento Nacional de Informações Penitenciárias - INFOPEN. Technical report, Ministério da Justiça.
- [DEPEN 2016] DEPEN (2016). Departamento Penitenciário Nacional (Acessado em Mar.2016). <http://dados.mj.gov.br/dataset/infopen-levantamento-nacional-de-informacoes-penitenciarias>.
- [Ding and Densham 1996] Ding, Y. and Densham, P. J. (1996). *Geographical Information Systems*, chapter Spatial Strategies for Parallel Spatial Modeling. Geographical Information Systems.
- [Ding et al. 1992] Ding, Y., Densham, P. J., and Armstrong, M. P. (1992). Parallel processing for network analysis: Decomposing shortest path algorithms for mimd computers. *5th International Symposium on Data Handling, Charleston, SC*.
- [FAA 2014] FAA (2014). Global positioning system (gps) standard positioning service (sps) performance analysis report. Technical report, Federal Aviation Administration – FAA.
- [Feinberg and Adrian 2015] Feinberg, D. and Adrian, M. (2015). The State of Open-Source RDBMSs, 2015 (Acessado em Mar.2016). <https://www.gartner.com/doc/3033819/state-opensource-rdbms->.
- [Flynn 1972] Flynn, M. J. (1972). Some computer organizations and their effectiveness. *IEEE transactions on computers*.
- [Francelino 2003] Francelino, M. R. (2003). Introdução ao geoprocessamento. *Caratinga: Centro*.
- [Handcock et al. 2009] Handcock, R. N., Swain, D. L., Bishop-Hurley, G. J., Patison, K. P., Wark, T., Valencia, P., Corke, P., and O’Neill, C. J. (2009). Monitoring animal behaviour and environmental interactions using wireless sensor networks, gps collars and satellite remote sensing. *Sensors*.
- [IBGE 2016] IBGE (2016). 7th United Nations Regional Cartographic Conference for the Americas, New Yourk, 22-26, 2001 (Acessado em Mar.2016). <http://http://www.ibge.gov.br/home/geociencias/geodesia/sirgas/onu2001.pdf>.
- [Jakkhupan and Klaypaksee 2014] Jakkhupan, W. and Klaypaksee, P. (2014). A web-based criminal record system using mobile device: A case study of hat yai municipality. In *Wireless and Mobile, 2014 IEEE Asia Pacific Conference on. IEEE*.
- [Ki and Hee 2011] Ki, Y. P. and Hee, Y. Y. (2011). Crime prevention system based on context-awareness. In *Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on. IEEE*.

- [Lewis et al. 2004] Lewis, F. L. et al. (2004). Wireless sensor networks. *Smart environments: technologies, protocols, and applications*.
- [Liu et al. 2012] Liu, Q., Deng, M., Shi, Y., and Wang, J. (2012). A density-based spatial clustering algorithm considering both spatial proximity and attribute similarity. *Computers & Geosciences*.
- [Louhichi et al. 2014] Louhichi, S., Gzara, M., and Ben Abdallah, H. (2014). A density based algorithm for discovering clusters with varied density. In *Computer Applications and Information Systems (WCCAIS), 2014 World Congress on*. IEEE.
- [Luo et al. 2002] Luo, R. C., Yih, C.-C., and Su, K. L. (2002). Multisensor fusion and integration: approaches, applications, and future research directions. *IEEE Sensors journal*.
- [Morreale and Suleski 2009] Morreale, P. and Suleski, R. (2009). System design and analysis of a web-based application for sensor network data integration and real-time presentation. In *Systems Conference, 2009 3rd Annual IEEE*. IEEE.
- [Obe and Hsu 2015] Obe, R. O. and Hsu, L. S. (2015). *PostGIS in action*. Manning Publications Co.
- [Oduor et al. 2014] Oduor, C., Acosta, F., and Makhanu, E. (2014). The adoption of mobile technology as a tool for situational crime prevention in kenya. In *IST-Africa Conference Proceedings, 2014*. IEEE.
- [OndeFuiRoubado 2016] OndeFuiRoubado (2016). Onde Fui Roubado). <http://http://www.ondefuiroubado.com.br/>.
- [OpenShaw and Turton 2014] OpenShaw, S. and Turton, I. (2014). *High-performance Computing and the Art of Parallel Programming*. Routledge.
- [OSGeo 2016] OSGeo (2016). Open Source Geospatial Foundation (Acessado em Mar.2016). <http://http://www.osgeo.org/>.
- [Peucker and Douglas 1975] Peucker, T. K. and Douglas, D. H. (1975). Detection of surface-specific points by local parallel processing of discrete terrain elevation data. *Computer Graphics and image processing*.
- [Pimenta 2016] Pimenta, I. L. (2016). Diretrizes para tratamento e proteção de dados na monitoração eletrônica de pessoas. Technical report, Ministério da Justiça.
- [Premebida 2005] Premebida, C. (2005). Seguimento e Associação de Dados em Tarefas de Robótica Móvel Usando Informação de um LRF. In *Technical Report No ISRLM2005/05*. Universidade de Coimbra.
- [Quinn 1986] Quinn, M. J. (1986). *Designing efficient algorithms for parallel computers*. McGraw-Hill, Inc.

- [Samet 1990] Samet, H. (1990). *The design and analysis of spatial data structures*. Addison-Wesley Reading, MA.
- [Sathyadevan et al. 2014] Sathyadevan, S., Devan, M., and Surya Gangadharan, S. (2014). Crime analysis and prediction using data mining. In *Networks & Soft Computing (ICNSC), 2014 First International Conference on*. IEEE.
- [Steinberg et al. 1999] Steinberg, A. N., Bowman, C. L., and White, F. E. (1999). Revisions to the JDL data fusion model. In *AeroSense'99*. International Society for Optics and Photonics.
- [Tomlin 1990] Tomlin, C. D. (1990). *Geographic information systems and cartographic modeling*. Prentice Hall.
- [Urbano and Dettki 2014] Urbano, F. and Dettki, H. (2014). *Spatial Database for GPS Wildlife Tracking Data*, chapter Storing tracking data in an advanced database platform (postgresql), pages 9–24. Springer.
- [Xia et al. 2012] Xia, Y., Liu, Y., Ye, Z., Wu, W., and Zhu, M. (2012). Quadtree-based domain decomposition for parallel map-matching on gps data. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE.