



Universidade Nova de Lisboa  
Faculdade de Ciências e Tecnologia  
Departamento de Informática

Dissertação de Mestrado em Engenharia Informática  
2º Semestre, 2008/2009

Pesquisa de Imagens de Rosto  
Nº 25951 Pedro Miguel Chambel Silva

Orientador

Prof<sup>a</sup>. Doutora Fernanda Maria Barquinha Tavares Vieira Barbosa

29 de Julho de 2009



Nº do aluno: 25951

Nome: Pedro Miguel Chambel Silva

Título da dissertação: Pesquisa de Imagens de Rosto

Palavras-Chave:

- Pesquisa por semelhança
- Imagens de rosto
- Estruturas de dados métricas
- Processamento de imagem

Keywords:

- Similarity search
- Face images
- Metric data structures
- Image processing



## **Agradecimentos**

Em primeiro lugar, gostaria de agradecer à Professora Fernanda Barbosa, pela orientação e apoio prestado em todos os momentos, pela sua disponibilidade para esclarecer as dúvidas e pelas sugestões ao longo de todo o trabalho.

Aos Professores Adriano Lopes, Nuno Correia, Margarida Mamede e Jorge Cruz pela ajuda, disponibilidade, e pelas sugestões e críticas, sempre construtivas, que permitiram enriquecer o meu trabalho.

Aos meus colegas Filipe Grangeiro e Alexandre Martins pela ajuda e troca de ideias.

Aos meus pais António e Maria Eugénia por todo o amor, compreensão e dedicação demonstrada ao longo de todos estes anos.

Aos meus avós Manuel e Idalina, e irmã Sílvia pelo apoio, amizade e carinho que sempre demonstraram.

A todos os meus amigos, principalmente ao Francisco, Silas, Nuno e Kátia pelo apoio e companheirismo.



## Resumo

Em muitas aplicações existe a necessidade de obter da base de dados o conjunto de imagens de rosto semelhantes a uma dada imagem de consulta.

Como forma de comparar imagens de rosto são usadas medidas de semelhança que permitem medir a similaridade entre duas imagens. A similaridade entre duas imagens é calculada com base nas características extraídas de cada imagem de rosto. No entanto, a alta dimensão das imagens de rosto leva a que a computação da medida de distância seja bastante pesada.

Em bases de dados com um grande volume de dados, a pesquisa de imagens de rosto semelhantes a uma dada imagem de consulta pode tornar-se impraticável, principalmente se for realizada de forma exaustiva. Uma forma de agilizar a pesquisa por semelhança em espaços métricos é usando estruturas de dados métricas.

Neste trabalho pretende-se avaliar a aplicabilidade e eficiência das estruturas de dados métricas na pesquisa por semelhança de imagens de rosto.

Este trabalho envolveu um estudo das representações de imagens de rosto e das funções de distância entre imagens de rosto de modo a avaliar a aplicabilidade das estruturas de dados métricas, e uma avaliação da pesquisa por alcance com 8 estruturas de dados métricas em 4 bases de dados sobre 4 espaços métricos.

Os resultados obtidos nesta avaliação permitem afirmar que as estruturas de dados métricas podem ser usadas neste domínio de aplicação e que na pesquisa por alcance a sua utilização minimiza significativamente o número de cálculos realizados face à pesquisa exaustiva, principalmente em bases de dados de grandes dimensões.



## **Abstract**

In many applications there's the need to get the set of face images from a database that are similar to a given face image.

In order to compare face images, similarity measures are used to measure the similarity between two face images. In this case, the facial features are extracted from each face image and are used to measure the similarity between face images.

However, the high dimensionality of face images leads to a heavy computation of the distance measure.

In large databases the search of face images that are similar to a face query can be impracticable, particularly in an exhaustive searching mode. One way to speed up the face similarity search in metric spaces is by using metric data structures.

The goal of this work is to evaluate the applicability and efficiency of metric data structures in the face similarity search.

In order to evaluate the applicability of metric data structures in this domain, this work involve a study of face images representations and distance measures to get the similarity between face images, and an evaluation of 8 metric data structures in 4 databases over 4 metric spaces.

Based on this study, we can conclude that the metric data structures can be used in this application domain and that the use of metric data structures in the range queries of similar face images can significantly decrease the number of distance computations when compared to the exhaustive search, especially in large databases.



## Índice

<b>Agradecimentos .....</b>	<b>5</b>
<b>Resumo .....</b>	<b>7</b>
<b>Abstract .....</b>	<b>9</b>
<b>Índice .....</b>	<b>11</b>
<b>Índice de figuras .....</b>	<b>17</b>
<b>Índice de tabelas .....</b>	<b>21</b>
<b>Capítulo 1. Introdução .....</b>	<b>23</b>
1.1. Motivação .....	23
1.2. Enquadramento .....	24
1.3. Objectivos .....	25
1.4. Metodologia .....	25
1.5. Contribuições .....	26
1.6. Estrutura do documento .....	26
<b>Parte I – Trabalho relacionado .....</b>	<b>29</b>
<b>Capítulo 2. Representação de imagens de rosto .....</b>	<b>31</b>
2.1. Classificação dos métodos de representação de rostos .....	33
2.1.1. Métodos baseados nas características globais .....	33
2.1.2. Métodos baseados nas características locais .....	33
2.1.3. Métodos híbridos .....	34
2.2. Técnicas de extracção de características faciais .....	34
2.2.1. <i>PCA – Principal Component Analysis</i> .....	35
2.2.2. <i>LDA – Linear Discriminant Analysis</i> .....	36
2.2.3. <i>SIFT – Scale Invariant Feature Transform</i> .....	37
2.3. Métodos relevantes para a representação da imagem de rosto .....	37
2.3.1. Método <i>Eigenfaces</i> .....	38
2.3.2. Método <i>Fisherfaces</i> .....	39

2.3.3. Técnica de <i>SIFT</i> aplicada às imagens de rosto .....	40
2.3.4. Análise comparativa .....	41
2.4. Conclusão e contributos .....	42
<b>Capítulo 3. Medidas de semelhança entre imagens de rosto.....</b>	<b>43</b>
3.1. Medidas de distância baseadas na métrica de <i>Minkowski</i> .....	44
3.1.1. Distância Euclidiana – $L_2$ .....	44
3.1.2. Distância de <i>Manhattan</i> – $L_1$ .....	45
3.1.3. Distância de <i>Chebyshev</i> - $L_\infty$ .....	46
3.2. Covariância.....	46
3.3. Correlação.....	47
3.4. <i>Mahalanobis</i> .....	48
3.4.1. <i>Mahalanobis</i> – $L_1$ .....	48
3.4.2. <i>Mahalanobis</i> – $L_2$ .....	48
3.4.3. <i>Mahalanobis</i> – Co-seno.....	49
3.4.4. Medida de <i>Yambor</i> .....	49
3.5. Distância de <i>Hausdorff</i> .....	50
3.5.1. Medida parcial de <i>Hausdorff</i> .....	51
3.5.1.1. Medida de <i>Hausdorff</i> dos níveis de cinzento .....	51
3.6. Análise comparativa .....	53
3.7. Conclusão e contributos .....	53
<b>Capítulo 4. Pesquisa por semelhança em espaços métricos .....</b>	<b>55</b>
4.1. Classificação das estruturas de dados métricas .....	56
4.2. Estruturas de dados métricas .....	60
4.2.1. <i>LC – List of Clusters</i> .....	61
4.2.1.1. Algoritmo de construção .....	62
4.2.1.2. Algoritmo de pesquisa por alcance.....	62
4.2.2. <i>RLC – Recursive Lists Of Clusters</i> .....	64
4.2.2.1. Algoritmo de construção .....	65
4.2.2.2. Algoritmo de inserção .....	65
4.2.2.3. Algoritmo de remoção .....	66
4.2.2.4. Algoritmo de pesquisa por alcance.....	67
4.2.3. <i>VPTree – Vantage Point Tree</i> .....	71
4.2.3.1. Algoritmo de construção .....	72
4.2.3.2. Algoritmo de pesquisa por alcance.....	73
4.2.4. <i>GNAT – Geometric Near-neighbor Access Tree</i> .....	74
4.2.4.1. Algoritmo de construção .....	75
4.2.4.2. Algoritmo de pesquisa por alcance.....	75

4.2.5. <i>LAESA - Linear Approximating Eliminating Search Algorithm</i> .....	77
4.2.5.1. Algoritmo de construção .....	77
4.2.5.2. Algoritmo de pesquisa por alcance.....	77
4.2.6. <i>DSAT – Dynamic Spatial Approximation Tree</i> .....	78
4.2.6.1. Algoritmo de construção .....	79
4.2.6.2. Algoritmo de inserção .....	79
4.2.6.3. Algoritmo de remoção .....	81
4.2.6.4. Algoritmo de pesquisa por alcance.....	82
4.2.7. <i>HDSAT – Hybrid Dynamic Spatial Approximation Tree</i> .....	83
4.2.7.1. <i>HDSAT1 – Hybrid Dynamic Spatial Approximation Tree 1</i> .....	84
4.2.7.2. <i>HDSAT2 – Hybrid Dynamic Spatial Approximation Tree 2</i> .....	85
4.3. Análise comparativa .....	86
4.4. Conclusão .....	87

## **Parte II – Pacote de estruturas de dados métricas..... 89**

### **Capítulo 5. Pacote de estruturas de dados métricas ..... 91**

5.1. Organização do pacote.....	91
5.2. Descrição das interfaces .....	92
5.3. Descrição das classes.....	94
5.3.1. <i>LAESA</i> .....	94
5.3.2. <i>VPTree</i> .....	95
5.3.3. <i>LC</i> .....	95
5.3.4. <i>GNAT</i> .....	96
5.3.5. <i>DSAT</i> e variantes .....	97
5.3.6. <i>RLC</i> .....	98
5.4. Conclusão .....	100

## **Parte III – Avaliações e protótipo..... 101**

### **Capítulo 6. Bases de dados utilizadas..... 103**

6.1. Base de dados <i>Faces94</i> .....	103
6.2. Base de dados <i>JAFFE</i> .....	104
6.3. Base de dados <i>AT&amp;T</i> .....	105
6.4. Base de dados <i>Yalefaces</i> .....	106
6.5. Conclusão .....	106

### **Capítulo 7. Espaços métricos ..... 109**

7.1. Implementação.....	109
7.1.1. Representação de imagens.....	109

7.1.2. Métricas .....	110
7.2. Método de avaliação .....	111
7.3. Avaliação realizada e resultados.....	113
7.3.1. Pesquisas realizadas.....	114
7.3.2. Resultados.....	115
7.4. Conclusão .....	117
<b>Capítulo 8. Pesquisa por alcance com estruturas de dados métricas.</b>	<b>119</b>
8.1. Dados a processar .....	120
8.1.1. Base de dados e conjuntos de consulta.....	120
8.1.2. Raios de pesquisa .....	120
8.2. Parametrização das estruturas de dados métricas .....	123
8.2.1. <i>LAESA</i> .....	123
8.2.2. <i>DSAT</i> e variantes .....	124
8.2.3. <i>GNAT</i> .....	124
8.2.4. <i>LC</i> .....	125
8.2.5. <i>RLC</i> .....	125
8.3. Avaliação das estruturas de dados .....	126
8.3.1. Raios de pesquisa utilizados na avaliação .....	127
8.4. Resultados da avaliação.....	128
8.4.1. Avaliação das estruturas de dados com a distância Euclidiana.....	128
8.4.2. Avaliação das estruturas de dados com a distância de <i>Manhattan</i> .....	130
8.4.3. Avaliação das estruturas de dados com a <i>Mahalanobis – L1</i> .....	133
8.4.4. Avaliação das estruturas de dados com a <i>Mahalanobis – L2</i> .....	135
8.5. Conclusão .....	137
<b>Capítulo 9. Protótipo.....</b>	<b>139</b>
9.1. Interface e funcionalidades .....	139
9.1.1. Pesquisa por alcance.....	140
9.1.2. Estatísticas .....	142
9.2. Arquitetura do sistema.....	143
9.2.1. Bases de dados utilizadas .....	145
9.2.2. Carregamento da informação.....	145
9.3. Conclusão .....	146
<b>Capítulo 10. Conclusão e trabalho futuro.....</b>	<b>147</b>
10.1. Conclusão .....	147
10.2. Trabalho futuro .....	149
<b>Bibliografia.....</b>	<b>151</b>

<b>Anexos.....</b>	<b>157</b>
A 1. Normalização da informação .....	157
A 1.1. Métodos baseados na normalização geométrica.....	157
A 1.1.1. <i>Funneling</i> .....	157
A 1.2. Métodos baseados na normalização fotométrica .....	158
A 1.2.1. <i>Homomorphic Filter</i> .....	158
A 1.2.2. <i>LogAbout</i> .....	159
A 1.2.3. <i>Histogram Equalization</i> .....	160
A 1.2.4. <i>Wavelets</i> .....	160
A 2. Medidas de distância aplicadas a imagens.....	163
A 2.1. <i>EMD – Earth Mover’s Distance</i> .....	163
A 3. Parametrização das estruturas de dados métricas .....	167
A 3.1. Parametrização da <i>LAESA</i> .....	167
A 3.2. Parametrização da <i>GNAT</i> .....	168
A 3.3. Parametrização da <i>DSAT</i> e variantes .....	169
A 3.4. Parametrização da <i>LC</i> .....	174
A 3.5. Parametrização da <i>RLC</i> .....	176



## Índice de figuras

Figura 2.1 – Esquema geral de um sistema de reconhecimento facial .....	31
Figura 2.2 – Exemplo de descritores extraídos com a técnica <i>SIFT</i> .....	40
Figura 4.1 – Partição do espaço com base em 2 <i>pivots</i> .....	57
Figura 4.2 – Descarte e selecção de elementos com <i>pivots</i> .....	58
Figura 4.3 – Partição do espaço em 4 agrupamentos .....	59
Figura 4.4 – Descarte e selecção de elementos num agrupamento .....	60
Figura 4.5 – Partição do espaço em agrupamentos na <i>LC</i> .....	61
Figura 4.6 – Construção da <i>LC</i> .....	62
Figura 4.7 – Pesquisa por alcance na <i>LC</i> .....	63
Figura 4.8 – Duas situações em que o espaço de pesquisa intersecta o agrupamento....	63
Figura 4.9 – Duas situações onde é necessário pesquisar nos agrupamentos seguintes. 64	
Figura 4.10 – Exemplo de uma <i>RLC</i> .....	65
Figura 4.11 – Inserção de um novo elemento na <i>RLC</i> .....	66
Figura 4.12 – Remoção de um elemento na <i>RLC</i> .....	67
Figura 4.13 – Pesquisa no agrupamento abrange o centro deste .....	68
Figura 4.14 – Pesquisa no agrupamento não abrange o centro deste .....	69
Figura 4.15 – Pesquisa dentro do agrupamento. O elemento $x$ é descartado sem calcular $d(q,x)$ .....	70
Figura 4.16 – Pesquisa dentro do agrupamento. O elemento $x$ é seleccionado sem calcular $d(q,x)$ .....	70
Figura 4.17 – Exemplo de uma <i>VPtree</i> .....	71
Figura 4.18 – Construção de uma <i>VPtree</i> .....	72
Figura 4.19 – Pesquisa por alcance numa <i>VPtree</i> .....	73
Figura 4.20 – Exemplo do primeiro nível de uma <i>GNAT</i> com 4 centros .....	74
Figura 4.21 – Exemplo da utilização da tabela de intervalos .....	76
Figura 4.22 – Exemplo de uma <i>DSAT</i> no espaço e da sua representação hierárquica ...	79
Figura 4.23 – Exemplo de uma inserção numa <i>DSAT</i> .....	80
Figura 4.24 – Exemplo de inserção na <i>DSAT</i> num nó que excedeu a capacidade .....	80
Figura 4.25 – Operação de pesquisa na <i>DSAT</i> .....	83
Figura 4.26 – Conjunto de <i>pivots</i> e respectivas distâncias do nó $u_2$ de uma <i>HDSAT1</i> ..	84
Figura 5.1 – Diagrama de classes do pacote “dataStructures” .....	92

Figura 5.2 – Interface <i>IMetricDataStructure&lt;E&gt;</i> .....	93
Figura 5.3 – Interface <i>IDynamicMetricDataStructure&lt;E&gt;</i> .....	93
Figura 5.4 – Interface <i>IMetricComparator&lt;E&gt;</i> .....	94
Figura 5.5 – Classe <i>LAESA</i> .....	94
Figura 5.6 – Classe <i>VPTree</i> .....	95
Figura 5.7 – Classe <i>LC</i> .....	96
Figura 5.8 – Classe <i>GNAT</i> .....	97
Figura 5.9 – Classes <i>DSAT</i> , <i>HDSAT1</i> e <i>HDSAT2</i> .....	98
Figura 5.10 – Classe <i>RLC</i> .....	99
Figura 6.1 – Amostras de imagens da base de dados <i>Faces94</i> .....	104
Figura 6.2 – Exemplo dos vários tipos de expressão da mesma pessoa na <i>JAFFE</i> .....	105
Figura 6.3 – Exemplo das diferentes variações em imagens associadas ao mesmo indivíduo na <i>AT&amp;T</i> .....	105
Figura 6.4 – Exemplo das diferentes variações em imagens associadas ao mesmo indivíduo na <i>Yalefaces</i> .....	106
Figura 7.1 – Diagrama de classes das várias métricas.....	111
Figura 7.2 – Avaliação da distância Euclidiana .....	115
Figura 7.3 – Avaliação da distância de <i>Manhattan</i> .....	115
Figura 7.4 – Avaliação da métrica <i>Mahalanobis – L1</i> .....	116
Figura 7.5 – Avaliação da métrica <i>Mahalanobis – L2</i> .....	116
Figura 7.6 – Gráfico <i>ROC</i> com a média das taxas de verdadeiros e falsos positivos das quatro bases de dados para cada métrica. ....	117
Figura 8.1 – Construção de 3 instâncias da <i>RLC</i> com a base de dados <i>Faces94</i> .....	126
Figura 8.2 – Média do nº de computações das várias estruturas de dados métricas nas diferentes bases de dados e com a distância Euclidiana.....	129
Figura 8.3 – Média do nº de computações das várias estruturas de dados métricas nas diferentes bases de dados e com a distância de <i>Manhattan</i> .....	132
Figura 8.4 – Média do nº de computações das várias estruturas de dados métricas nas diferentes de dados e com a <i>Mahalalanobis – L1</i> .....	134
Figura 8.5 – Média do nº de computações das várias estruturas de dados métricas nas diferentes bases de dados e com a <i>Mahalalanobis – L2</i> .....	136
Figura 9.1 – Descrição dos componentes da interface .....	140
Figura 9.2 – Exemplo da execução de uma pesquisa .....	142
Figura 9.3 – Diagrama de pacotes do protótipo.....	143
Figura A 1.1 – Esquema do funcionamento do <i>Homomorphic Filter</i> .....	158
Figura A 1.2 – Normalização com o método <i>LogAbout</i> .....	159
Figura A 1.3 – Aplicação do <i>Histogram Equalization</i> sobre a primeira imagem .....	160
Figura A 3.1 – Parametrização da <i>LAESA</i> .....	168

Figura A 3.2 – Comportamento da <i>GNAT</i> aquando da variação do nº de centros .....	169
Figura A 3.3 – Parametrização da <i>DSAT</i> e variantes na <i>Faces94</i> .....	170
Figura A 3.4 – Parametrização da <i>DSAT</i> e variantes na <i>JAFFE</i> .....	171
Figura A 3.5 – Parametrização da <i>DSAT</i> e variantes na <i>AT&amp;T</i> .....	172
Figura A 3.6 – Parametrização da <i>DSAT</i> e variantes na <i>Yalefaces</i> .....	173
Figura A 3.7 – Comportamento da <i>LC</i> com agrupamentos de raio fixo na <i>Faces94</i> ...	174
Figura A 3.8 – Comportamento da <i>LC</i> com agrupamentos de raio fixo na <i>JAFFE</i> .....	175
Figura A 3.9 – Comportamento da <i>LC</i> com agrupamentos de raio fixo na <i>AT&amp;T</i> .....	175
Figura A 3.10 – Comportamento da <i>LC</i> com agrupamentos de raio fixo na <i>Yalefaces</i>	176
Figura A 3.11 – Comportamento da <i>RLC</i> na <i>Faces94</i> .....	177
Figura A 3.12 – Comportamento da <i>RLC</i> na <i>JAFFE</i> .....	178
Figura A 3.13 – Comportamento da <i>RLC</i> na <i>AT&amp;T</i> .....	179
Figura A 3.14 – Comportamento da <i>RLC</i> na <i>Yalefaces</i> .....	180



## Índice de tabelas

Tabela 2.1 – Sensibilidade dos métodos baseados em características globais e locais a variações de diversas condições .....	34
Tabela 6.1 – Número total de indivíduos e de imagens de rosto nas diferentes bases de dados.....	107
Tabela 7.1 – N° de imagens de rosto seleccionadas em cada base de dados.....	114
Tabela 8.1 – Raios de pesquisa utilizados na parametrização das estruturas e respectivas médias das taxas de verdadeiros e falsos positivos e do n° de resultados obtidos das várias pesquisas .....	122
Tabela 8.2 – Número de <i>pivots</i> na <i>LAESA</i> .....	123
Tabela 8.3 – Aridade máxima na <i>DSAT</i> , <i>HDSAT1</i> e <i>HDSAT2</i> .....	124
Tabela 8.4 – Raios de construção da <i>LC</i> .....	125
Tabela 8.5 – Raio de cada agrupamento da <i>RLC</i> .....	125
Tabela 8.6 – Capacidade do vector na <i>RLC</i> .....	125
Tabela 8.7 – Raios de pesquisa utilizados na avaliação das pesquisas por alcance .....	127
Tabela 8.8 – Média do n° de elementos obtidos com os diferentes raios associados a cada uma das métricas .....	127
Tabela 8.9 – Percentagem da média do n° de computações efectuadas em relação ao tamanho da base de dados com a distância Euclidiana .....	128
Tabela 8.10 – Percentagem da média do n° de computações efectuadas em relação ao tamanho da base de dados com a distância de <i>Manhattan</i> .....	131
Tabela 8.11 – Percentagem da média do n° de computações efectuadas em relação ao tamanho da base de dados com a <i>Mahalanobis – L1</i> .....	133
Tabela 8.12 – Percentagem da média do n° de computações efectuadas em relação ao tamanho da base de dados com a <i>Mahalanobis – L2</i> .....	135
Tabela 9.1 – Descrição das classes mais relevantes do pacote “frEigenfaces” .....	144



## Capítulo 1. Introdução

O tema central desta tese é a avaliação das estruturas de dados métricas. Em particular, esta tese faz essa avaliação na pesquisa por alcance no domínio das imagens de rosto, com diferentes estruturas de dados métricas. O trabalho realizado, no contexto desta avaliação, não está contudo limitado à avaliação de um conjunto de pesquisas por alcance nas diferentes estruturas de dados, mas também inclui um estudo da definição e avaliação de espaços métricos neste domínio de aplicação e um estudo das principais estruturas de dados métricas, como ficará claro durante a sua apresentação.

### 1.1. Motivação

O reconhecimento de imagens de rosto tem recebido muita atenção nos últimos tempos, devido ao vasto número de aplicações em diferentes áreas em que pode ser usado tais como controlo de acesso, vigilância, entretenimento, identificação criminal, entre outras [1].

A ideia fundamental deste tipo de sistemas consiste em identificar ou reconhecer um determinado indivíduo num determinado cenário, tendo por base uma colecção de imagens de rosto armazenadas [1]. Para este tipo de sistemas, as operações tradicionais em bases de dados, como a pesquisa exacta de elementos, são raras. Desta forma, estamos interessados em pesquisas aproximadas que consistem na pesquisa de elementos que são similares ou próximos a uma determinada consulta, neste caso de rostos.

As pesquisas aproximadas são classificadas em pesquisa por alcance (termo em inglês, *range query*) e pesquisa dos  $k$  vizinhos mais próximos (termo em inglês, *k-nearest neighbor query*) [25]. Na pesquisa por alcance o objectivo consiste em encontrar o conjunto de elementos da base de dados cuja distância em relação ao elemento de consulta não excede um determinado raio de pesquisa. Em relação à

pesquisa dos  $k$  vizinhos mais próximos, o objectivo consiste em encontrar os  $k$  elementos da base de dados mais similares em relação ao elemento de consulta.

Para realizar estas pesquisas são usadas medidas de semelhança que permitem medir a similaridade/proximidade entre as imagens. No entanto, em bases de dados de grandes dimensões, a procura exaustiva de imagens de rosto similares, a uma determinada imagem de rosto de consulta, é um processo computacionalmente pesado.

Este facto resulta, não só, do volume de dados, como também do facto que muitas vezes o cálculo da semelhança entre elementos é muito pesado.

Logo, é essencial estudar e analisar técnicas que agilizem o processo de pesquisa por similaridade. Uma dessas técnicas baseia-se na noção matemática de espaço métrico, o qual é um conceito muito usado quando é necessário medir a proximidade dos elementos. Este conceito permite a utilização de técnicas que minimizam o número de cálculos de similaridade entre os vários objectos da base de dados. Com base no conceito de espaço métrico, foram propostas diferentes estruturas de dados que particionam o espaço com base na similaridade/proximidade entre elementos, denominadas de estruturas de dados métricas [25], com o objectivo de minimizar o número de cálculos da função de distância entre objectos da base de dados aquando de uma pesquisa.

## 1.2. Enquadramento

Dado o grande volume de informação em diferentes domínios de aplicação e a importância de realizar eficientemente pesquisas aproximadas nessas aplicações, é fundamental avaliar a aplicabilidade e eficiência de técnicas baseadas no conceito de espaço métrico, nomeadamente as estruturas de dados métricas, em diferentes domínios de aplicação. Assim sendo, esta tese faz parte de um conjunto formado por três trabalhos de investigação, na área de Recuperação de Informação (termo em inglês, *Information Retrieval*), que pretendem avaliar as estruturas de dados métricas em diferentes domínios de aplicação nomeadamente música, imagens de rosto e eventos geográficos [19].

Todos estes trabalhos foram motivados por um dos objectivos da equipa *ADSIR* (*Algorithms and Data Structures for Information Retrieval*) do *CITI* (*Center for Informatics and Information Technologies*), o qual pretende avaliar a estrutura de dados

métrica *RLC* [41][6], desenvolvida pela professora Margarida Mamede. É de salientar que já existem trabalhos semelhantes em outros domínios de aplicação [7][40][42].

### **1.3. Objectivos**

O objectivo desta dissertação é estudar a aplicabilidade e eficiência das estruturas de dados métricas na pesquisa por alcance de imagens de rosto. Para isso, foi criado um protótipo que permite realizar e avaliar pesquisas de imagens de rosto semelhantes, utilizando diferentes estruturas de dados métricas.

Este estudo envolve 4 bases de dados de imagens de rosto, já utilizadas em trabalhos de investigação relacionados, as quais serão definidas sobre 4 espaços métricos. As pesquisas por alcance realizadas na avaliação foram efectuadas sobre estas bases de dados, utilizando 8 estruturas de dados métricas. A avaliação de eficiência das pesquisas é realizada com base no número de cálculos de distância realizados.

### **1.4. Metodologia**

De modo a alcançar o objectivo pretendido, foram realizadas as seguintes actividades:

- Pesquisa bibliográfica dos trabalhos existentes no domínio de imagens de rosto;
- Estudo dos diferentes espaços métricos que podem ser definidos no domínio de imagens de rosto. Este estudo envolve dois aspectos:
  1. Possíveis representações para imagens de rosto;
  2. Critérios de semelhança existentes, usados para a comparação de imagens de rosto;
- Estudo das estruturas de dados métricas existentes que agilizam o processo de pesquisas por proximidade em espaços métricos;
- Avaliação e implementação dos espaços métricos definidos;
- Implementação das estruturas de dados métricas envolvidas na avaliação;
- Avaliação das pesquisas por alcance nas diferentes estruturas de dados métricas implementadas.

De modo a tornar público este trabalho, foram realizadas as seguintes actividades:

- Escrita de 2 artigos [19][43];
- Página que descreve o estudo realizado neste trabalho, disponível em [44].

## **1.5. Contribuições**

Este trabalho de investigação pretende contribuir para agilizar o processo de pesquisas em imagens de rosto, principalmente no que diz respeito à eficiência temporal, utilizando estruturas de dados métricas. Para além disso, pretende-se realizar:

- Um estudo dos espaços métricos existentes, o qual envolve a análise das possíveis representações e funções de semelhança das imagens de rosto;
- Uma avaliação das diferentes estruturas de dados métricas nas pesquisas por alcance em imagens de rosto.

Para além das contribuições inerentes a esta avaliação da aplicabilidade e eficiência das estruturas de dados métricas em imagens de rosto, neste trabalho pretende-se implementar um protótipo que permite fazer pesquisas por alcance em imagens de rosto.

## **1.6. Estrutura do documento**

Este documento está estruturado em 3 partes compostas por 10 capítulos. Neste primeiro capítulo é feita uma apresentação do tema, nomeadamente no que diz respeito ao seu enquadramento e motivação e são definidos os objectivos, metodologia e contribuições previstas. Após o capítulo 1, a parte I sintetiza os aspectos/técnicas existentes dos principais tópicos de investigação em que esta tese se insere, os quais dizem respeito à definição de espaços métricos (representação e função de semelhança) em imagens de rosto e estruturas de dados métricas. Esta parte é composta por 3 capítulos:

- Representação de imagens de rosto: Apresenta um estudo das várias formas de representação de imagens de rosto que inclui as várias técnicas de extracção das características faciais e de redução da dimensão dos dados.
- Medidas de semelhança entre imagens de rosto: Apresenta um estudo das várias medidas de semelhança (e variantes destas) usadas para medir a

similaridade entre imagens de rosto.

- Pesquisa por semelhança em espaços métricos: Apresenta um estudo das várias estruturas de dados métricas existentes propostas com o objectivo de tornar as pesquisas por alcance mais eficientes, tendo como base a noção de espaço métrico.

Em cada um destes capítulos, que compõem a parte I, é apresentada uma secção que descreve uma conclusão acerca das técnicas enumeradas e indica qual o contributo dessas técnicas para o trabalho realizado.

A parte II é composta por um único capítulo e apresenta o pacote de estruturas de dados métricas implementadas nesta dissertação.

A parte III apresenta o protótipo desenvolvido e as avaliações. Esta parte é composta por 4 capítulos:

- Bases de dados: Apresenta uma descrição das várias bases de dados utilizadas na avaliação dos espaços métricos e das estruturas de dados métricas.
- Espaços métricos: Apresenta os espaços métricos usados nesta dissertação e a sua avaliação.
- Pesquisa por alcance com estruturas de dados métricas: Neste capítulo é realizada uma avaliação das várias estruturas de dados métricas com os espaços métricos implementados.
- Protótipo: Apresenta uma descrição do protótipo final.

Por último, o capítulo 10 apresenta as conclusões, contributos e identifica as direcções de trabalho futuro.



## **Parte I – Trabalho relacionado**



## Capítulo 2. Representação de imagens de rosto

Um dos aspectos fundamentais na pesquisa de elementos semelhantes é a representação dos dados, já que esta condiciona os critérios de semelhança e as técnicas utilizadas na procura.

De modo a poder definir o espaço de procura, que pretendemos que seja métrico, foram estudadas diferentes representações de imagens de face, utilizadas nos trabalhos mais relevantes na área de recuperação de informação em imagens de rosto.

A maior parte dos trabalhos existentes dizem respeito a sistemas de reconhecimento facial. De uma forma geral, estes sistemas envolvem as fases de detecção do rosto, extracção de características e reconhecimento, tal como podemos observar na figura 2.1 retirada de [1].

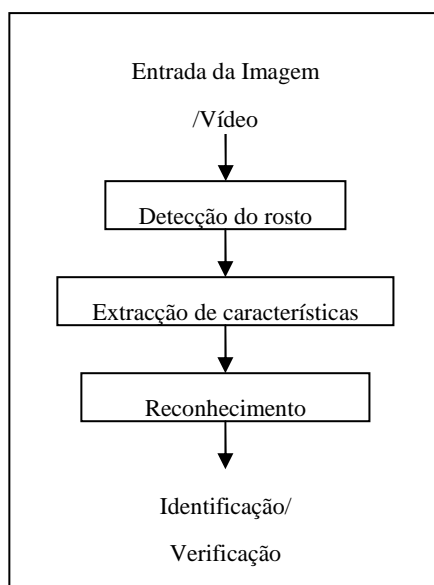


Figura 2.1 – Esquema geral de um sistema de reconhecimento facial

O objectivo da fase de detecção consiste em descobrir a presença de um rosto numa determinada imagem. Após a detecção desta, as imagens podem ser acompanhadas por um processo de normalização. A normalização consiste na preparação da informação

para que possa ser tratada nos passos seguintes. O objectivo, desta fase, consiste na transformação dos dados de forma a compensar as variações existentes, de modo a ter um melhoramento na taxa de reconhecimento. Desta forma, são tratados vários factores que condicionam o reconhecimento facial, tais como a variação das condições de iluminação, diferentes escalas e rotação dos rostos. Para uma breve descrição das técnicas de normalização ver anexo A1.

Depois da fase de detecção do rosto, é necessário realizar a tarefa de extracção das características que serão usadas posteriormente na classificação. Neste caso, verificamos uma redução da dimensão dos dados na medida em que as características extraídas correspondem a um subconjunto dos dados originais e permitem representar os mesmos. As fases de detecção e extracção de características não são necessariamente independentes e, desta forma, podem ser realizadas em simultâneo.

A última fase corresponde ao reconhecimento e envolve os cenários de identificação ou verificação. Existem dois conceitos que são importantes distinguir que correspondem a identificação e verificação. No primeiro a imagem de rosto de consulta é desconhecida e o sistema determina a identidade a partir de uma base de dados de indivíduos. Neste caso, a imagem de pesquisa é comparada com um conjunto de imagens de rostos de indivíduos conhecidos. Em relação à verificação, esta envolve a necessidade de autenticar uma determinada identidade, ou seja, o sistema necessita de confirmar ou rejeitar a identidade reivindicada.

Em todos estes trabalhos existem três aproximações para o reconhecimento facial: os métodos baseados nas características globais, os métodos baseados nas características locais, e os métodos híbridos.

Este capítulo descreve as aproximações, os métodos e as técnicas utilizadas na representação de imagens de rosto. Na secção 1 são definidas as aproximações existentes na representação de imagens de rosto. Na secção 2 são estudadas as técnicas usadas para a extracção das características associadas às imagens, enquanto que, na secção 3 são definidos alguns métodos relevantes na representação de imagens de rosto. Por último, na secção 4 são descritas as conclusões e os contributos destas técnicas nesta dissertação.

## **2.1. Classificação dos métodos de representação de rostos**

De acordo com as características extraídas do rosto, podemos classificar os métodos de reconhecimento facial em globais, locais e híbridos [1].

### **2.1.1. Métodos baseados nas características globais**

Os métodos baseados nas características globais utilizam toda a região do rosto para obter uma representação da imagem. A vantagem destes métodos está no facto de preservarem a informação detalhada da textura e forma, pois estas são úteis na distinção de rostos. Neste caso, como são utilizadas as informações globais do rosto, estes métodos têm como desvantagem a alta dimensão dos dados que, por sua vez, aumenta o custo computacional. Para contornar este problema são usadas várias técnicas de extracção de características que permitem reduzir a dimensão dos dados. Uma das técnicas com mais sucesso é o *PCA – Principal Component Analysis*, o qual permite a extracção das características mais significativas do rosto (ver secção 2.2.1). Além do *PCA*, existem também outras técnicas relevantes tais como *LDA – Linear Discriminant Analysis* [20] e o *SIFT – Scale Invariant Feature Transform* [22] (ver secções 2.2.2 e 2.2.3, respectivamente).

### **2.1.2. Métodos baseados nas características locais**

Nos métodos baseados nas características locais os elementos característicos do rosto tais como olhos, nariz e boca são primeiro extraídos e as suas posições e estatísticas locais são guardados de modo a manter a relação entre os diversos elementos. Desta forma, um rosto é representado como uma estrutura que engloba vários componentes e as diversas relações entre estes, os quais podem ser usados para distinguir as diferentes faces. Neste caso, as relações podem ser, por exemplo, as distâncias entre pontos que caracterizam o rosto tais como os olhos, tamanho do nariz, tamanho da testa, entre outras. Um exemplo baseado nas características locais foi utilizado em [22].

### 2.1.3. Métodos híbridos

A ideia dos métodos híbridos é baseada em estudos que indicam que o sistema de percepção humano usa tanto as características locais como globais para efectuar o reconhecimento facial. Motivados por esta ideia, os métodos híbridos são uma combinação das duas aproximações descritas anteriormente. Na tabela 2.1, retirada de [3], é feita uma comparação qualitativa entre os métodos globais e locais em relação à sensibilidade em diferentes variações de condições.

Desta forma, nos métodos híbridos são exploradas as diferentes propriedades das duas aproximações e é feita uma combinação das melhores características definidas por cada uma delas de modo a complementar a informação.

Variações	Métodos Locais	Métodos Globais
Pequenas variações	Não é sensível	Sensível
Grandes variações	Sensível	Muito sensível
Iluminação	Muito Sensível	Sensível
Expressão	Não é sensível	Sensível
Pose	Sensível	Muito sensível
Noise	Muito Sensível	Sensível
Occlusion	Sensível	Muito sensível

Tabela 2.1 – Sensibilidade dos métodos baseados em características globais e locais a variações de diversas condições

Não foram encontrados trabalhos relevantes que usem métodos híbridos.

## 2.2. Técnicas de extracção de características faciais

Nesta secção serão abordadas várias ferramentas e técnicas de extracção das características de imagens de rosto.

Nos métodos baseados nas características globais, verificamos que na representação das imagens de rosto é utilizada toda a região do rosto. Desta forma, as imagens de rostos com dimensão  $A \times L$  são usualmente representadas por vectores unidimensionais com dimensão  $AxL$  com a informação de cada *pixel*.

No entanto, verificamos que esta forma de representar as imagens apresenta informação redundante para o processo de reconhecimento. Outro aspecto importante, a ter em conta, está relacionado com o facto de esta forma de representação da informação apresentar uma dimensão muito grande para permitir um reconhecimento facial robusto e rápido. Assim, de modo representar apenas a informação mais relevante para o processo de reconhecimento, são usadas várias técnicas que permitem reduzir a dimensão dos dados tais como *PCA – Principal Component Analysis* [20], e *LDA – Linear Discriminant Analysis* [20][21].

O *SIFT – Scale Invariant Feature Transform* é outra técnica de extracção de características bastante usada para o reconhecimento de objectos, e recentemente existem alguns trabalhos [22] em que aplicaram o *SIFT* ao reconhecimento facial.

### **2.2.1. PCA – Principal Component Analysis**

O *PCA – Principal Component Analysis* é uma técnica estatística bastante utilizada para analisar dados e verificar como estes se relacionam. O principal objectivo consiste em encontrar uma transformação mais representativa e compacta dos dados. Desta forma, o *PCA* pode ser utilizado para projectar o espaço original de variáveis num espaço de dimensão mais reduzida. Assim, verificamos que as variáveis originais são reescritas em novas variáveis através de uma transformação para um novo sistema de coordenadas. Neste caso, as variáveis derivadas das originais são designadas de componentes principais. Estes correspondem aos chamados vectores próprios (termo em inglês, *eigenvectors*) da matriz de covariância obtida do conjunto de dados. Assim, cada componente principal é uma combinação linear de todas as variáveis originais.

Em relação às variáveis originais verificamos que estas têm a mesma importância estatística. No entanto, o mesmo não acontece nas componentes principais, e estas encontram-se ordenadas pela importância que representam. Desta forma, os primeiros componentes são os mais importantes na medida em que apresentam maior variância dos dados e representam os aspectos mais importantes destes. Neste caso, verificamos que grande parte da variância dos dados é representada por um número reduzido de componentes e, desta forma, é possível descartar os restantes sem grande perda de informação permitindo, deste modo, a redução da dimensão do espaço.

Turk e Pentland tornaram popular a técnica *PCA*, no reconhecimento de rostos, através do método designado de *Eigenfaces* [2]. Este corresponde à aplicação directa da técnica do *PCA* em imagens de rosto e será brevemente descrito na secção 2.3.1.

### **2.2.2. LDA – Linear Discriminant Analysis**

No conjunto de dados é possível formar classes de elementos. No entanto, no *PCA* verificamos que a informação relativa a cada classe não é levada em consideração, ou seja, os dados pertencentes à mesma classe (ou grupo) e de diferentes classes são tratados da mesma forma. Ao contrário do *PCA*, a técnica estatística *LDA – Linear Discriminant Analysis*, também conhecida como *FDA – Fisher Discriminant Analysis*, procura explorar a informação referente a cada classe. Neste caso, procura-se maximizar a variação entre classes e minimizar a variação dentro de classes com base em observações previamente classificadas nas diversas classes. Desta forma, o *LDA* possibilita, assim, a discriminação e classificação de objectos em classes mutuamente exclusivas com base nas características mensuráveis destes.

Assim, o objectivo principal da técnica *LDA* consiste em reduzir a dimensão dos dados e, ao mesmo tempo, preservar a informação discriminatória entre classes. Para tal, do conjunto de classes são criadas as matrizes de dispersão entre-classes (termo em inglês, *between-class scatter matrix*) e intra-classes (termo em inglês, *within-class scatter matrix*). Com base nas matrizes anteriores são obtidos os vectores próprios que maximizam a variação entre classes enquanto minimizam a variação intra-classes. Neste caso, a matriz intra-classes tem de ser não-singular (ou seja, invertível).

Em [20] é feita uma avaliação comparativa entre um sistema de reconhecimento facial baseado em *PCA* e outro em *LDA*. Os resultados dos testes efectuados mostraram que, quando o número de imagens de treino é pequeno, o sistema baseado em *PCA* pode superar o *LDA* em termos de desempenho no reconhecimento.

Um método representativo da aplicação do *LDA* no reconhecimento facial corresponde ao designado de *Fisherfaces* [21] e será brevemente descrito na secção 2.3.2.

### **2.2.3. SIFT – Scale Invariant Feature Transform**

O *SIFT – Scale Invariant Feature Transform* corresponde a uma técnica que permite extrair as características distintivas de imagens, sendo bastante usado para o reconhecimento e classificação de objectos em diferentes pontos de vista. Neste caso, verificamos que as características extraídas são invariantes em relação à escala e orientação da imagem.

O processo de extracção de características passa por quatro fases. A primeira corresponde à fase onde são detectadas as localizações de possíveis pontos de interesse invariantes à escala e orientação.

Na segunda, os pontos candidatos detectados anteriormente são examinados com maior detalhe e filtrados. Neste caso, é analisada com mais detalhe a informação dos vizinhos e um modelo é ajustado de modo a determinar correctamente a localização e escala. Com esta informação são descartados os pontos com baixo contraste e que desta forma são sensíveis ao ruído.

Na fase seguinte é definida uma orientação a cada ponto-chave. Neste caso, todas as propriedades do ponto-chave são medidas relativamente à orientação deste, possibilitando, assim, que estas sejam invariantes em relação à rotação.

O último passo consiste no cálculo de um descritor associado a cada um dos pontos-chave. Este descritor corresponde a um vector de tamanho fixo que é normalizado de forma a ser invariante em relação a mudanças de iluminação.

Assim, com base nos descritores extraídos, é possível realizar a correspondência entre duas imagens. Em [22] é explorado o uso do *SIFT* no contexto do reconhecimento facial.

## **2.3. Métodos relevantes para a representação da imagem de rosto**

Nesta secção serão abordados alguns métodos relevantes para a representação de imagens de rosto tais como o método de *Eigenfaces*, o método de *Fisherfaces*, e a técnica de *SIFT* aplicada a imagens de face.

### 2.3.1. Método *Eigenfaces*

O método designado por *Eigenfaces*, baseado nas características globais, foi apresentado por Turk e Pentland em 1991 [2] e foi motivado pelos trabalhos desenvolvidos por Sirovich e Kirby.

Neste método, cada imagem de rosto com largura  $D_x$  e altura  $D_y$ , em *pixels*, é inicialmente representada por um vector unidimensional de inteiros de dimensão  $D$ , em que  $D = D_x \times D_y$  e onde cada elemento desta matriz representa um *pixel* da imagem.

A ideia principal deste método consiste em reduzir a dimensão destes dados e, desta forma, executar o reconhecimento de rostos num espaço de menor dimensão. Desta forma, as imagens de rosto são projectadas no espaço de rostos (termo em inglês, *feature space*) que melhor descrevem a variação em relação às imagens conhecidas da base de dados.

Para tal, é necessário extrair as características principais do conjunto de treino. Estas designam-se de vectores próprios (termo em inglês, *eigenvectors*) e são extraídas por meio da ferramenta matemática designada de *PCA – Principal Component Analysis*. No contexto do reconhecimento facial, estes vectores próprios designam-se pelo termo em inglês de *eigenfaces*.

Os vectores próprios podem ser vistos como um conjunto de características (termo em inglês, *features*) que juntos conseguem caracterizar a variação entre as imagens de rosto. O número máximo de vectores próprios que pode ser usado é igual ao número de imagens do conjunto de treino. No entanto, o espaço de rostos pode ser aproximado com um número menor de vectores próprios tendo como principal vantagem uma maior eficiência computacional. Esta é verificada porque cada imagem de rosto é representada no espaço de rostos por um vector com a dimensão igual ao número de vectores próprios usado. Assim, ao reduzir o tamanho deste vector, são realizados menos cálculos na computação da medida de distância entre imagens de rosto. Desta forma, devem ser escolhidos os melhores vectores próprios. Associado a cada vector próprio existe a noção de valor próprio (termo em inglês, *eigenvalue*) que representa a contribuição do primeiro na representação das características principais. Desta forma, os melhores vectores próprios são os que têm os maiores valores próprios.

Para projectar uma imagem de rosto no espaço de rostos é necessário calcular o vector de pesos associado a esta. Para isso, são usados os vectores próprios calculados

anteriormente. Os pesos são obtidos com os vectores próprios do conjunto de treino que linearmente aproximam o rosto ou que podem ser usados para o reconstruir. Assim, um rosto é representado no espaço de rostos como um vector de pesos que representa a diferença em relação às características principais presentes nas imagens de rosto do conjunto de treino.

Desta forma, faz-se uma redução da dimensão dos dados na medida em que as imagens originais passam a ser representadas por vectores de pesos com dimensão igual ao número de vectores próprios usados.

Neste método, a identificação de uma imagem de rosto pode ser realizada procurando na base de dados pelo rosto com o vector de pesos mais parecido ao da imagem de teste, ou seja, com a menor distância. Para medir a similaridade entre dois vectores é usada a distância Euclidiana (ver capítulo 3 secção 3.1.1).

### **2.3.2. Método *Fisherfaces***

O método de reconhecimento facial designado por *Fisherfaces* referido em [21] foi apresentado por P. Belhumeur, J. Hespanha e D. Kriegman e é baseado na aplicação da técnica *LDA*.

Neste caso, o conjunto de imagens de rosto de treino é formado por várias classes. Assim, cada classe representa uma identidade de uma pessoa e, com base num conjunto de classes previamente conhecidas, o problema do reconhecimento pode ser formulado de modo a determinar a que classe pertence uma determinada pessoa desconhecida.

Como em geral o número de imagens de treino é muito menor que o número de *pixels* de cada imagem, a técnica *LDA* não pode ser aplicada directamente para o processo de reconhecimento pois, neste caso, a matriz intra-classes é singular. De modo a contornar este problema, é feita a projecção do conjunto de imagens de treino para um espaço de menor dimensão usando o *PCA*. Assim, a matriz intra-classes passa a ser não-singular pois a dimensão do espaço é reduzida para  $N-c$ , sendo  $N$  o número de imagens do conjunto de treino e  $c$  o número de classes.

De seguida, é aplicado a técnica *LDA* sendo a dimensão dos dados reduzida para  $c-1$ . Neste caso, os vectores próprios obtidos são usados para projectar as imagens de rostos num espaço de menor dimensão. O reconhecimento de uma imagem de rosto de teste é realizado projectando esta com base nos vectores próprios criados e utilizando

uma medida de semelhança para efectuar a comparação com as outras imagens de rosto projectadas no mesmo espaço.

### 2.3.3. Técnica de *SIFT* aplicada às imagens de rosto

Em [22] foram apresentadas três formas de reconhecimento facial usando o *SIFT* para realizar a extracção das características mais distintivas das imagens de rosto. Com esta técnica são extraídos vários ponto-chave de cada imagem de rosto. Cada ponto-chave contém a sua localização, orientação, escala e descritor.

Na figura 2.2 retirada de [22] está representado o conjunto de descritores extraídos duma imagem de rosto, com a técnica *SIFT*.



Figura 2.2 – Exemplo de descritores extraídos com a técnica *SIFT*

A diferença entre as três formas de reconhecimento está apenas na organização dos vários descritores extraídos com o *SIFT* e nas regiões do rosto consideradas para a extracção destes, ou seja, na representação da imagem.

Na primeira forma de representação, é aplicada a técnica do *SIFT* a toda a região facial, e cada rosto é representado pelos vários pontos-chave extraídos, em que cada ponto descreve a informação local do rosto. Neste caso, a correspondência entre os vários pontos-chave de cada imagem de rosto é realizada por meio de uma função que mede a similaridade entre os descritores (ver capítulo 3, secção 3.1.1). Aqui não é utilizada a informação relacionada com a localização, escala e orientação de cada ponto-chave.

Nesta forma de representação, pode acontecer que sejam comparados descritores associados a pontos de zonas diferentes da cara. Desta forma, na segunda forma de

representação trata-se de minimizar este problema, pelo que só se analisa a região dos olhos e da boca. Neste caso, é assumido que a posição dos olhos e da boca é previamente conhecida. Esta representação é motivada também pelo facto da maioria da informação relevante do rosto se encontrar nestas duas zonas. Para obter a representação de cada imagem de rosto são extraídas duas sub-imagens sendo a primeira localizada na zona dos olhos e a segunda na zona da boca. Assim, para realizar a correspondência entre imagens são comparadas as sub-imagens que estejam relacionadas, ou seja, olhos com olhos e boca com boca. Desta forma, é aplicada a técnica do primeiro método às sub-imagens relacionadas. O resultado final corresponde à média das duas medidas realizadas.

Em relação ao terceiro método de representação, podemos verificar que é semelhante ao anterior na medida em que a imagem também é dividida em várias sub-imagens. Neste caso, verificamos que a imagem original é dividida em várias sub-imagens parcialmente sobrepostas formando uma grelha de sub-imagens. Assim, este método funciona como o anterior pois apenas são comparadas as sub-imagens relacionadas pela mesma zona do rosto. Neste caso, o resultado final corresponde à média das distâncias que descrevem a similaridade das várias zonas.

#### **2.3.4. Análise comparativa**

O único trabalho que conseguimos na nossa pesquisa bibliográfica que apresenta uma análise comparativa dos 3 métodos descritos nas 3 secções anteriores foi o trabalho de reconhecimento facial baseado na técnica *SIFT* [37], o qual compara esta técnica com os métodos designados de *Eigenfaces* [2] e *Fisherfaces* [21]. Neste trabalho o método baseado na técnica *SIFT* foi o que obteve os melhores resultados quando comparado com o método designado de *Eigenfaces* e *Fisherfaces*. As bases de dados utilizadas foram a *AT&T* [35] e *Yalefaces* [36].

Na primeira base de dados, a proporção de resultados verdadeiros, com o método baseado na técnica *SIFT*, foi de 96.3%, enquanto que os métodos de *Eigenfaces* obtiveram, respectivamente, 92.9% e 93.8%.

Na segunda base de dados, a proporção de resultados verdadeiros, utilizando o método baseado na técnica *SIFT*, foi de 91.7% enquanto que os métodos de *Eigenfaces* e *Fisherfaces* obtiveram, respectivamente, 72.1% e 86.0%.

## 2.4. Conclusão e contributos

Neste capítulo foram descritos 3 métodos para a representação de imagens de rosto: *Eigenfaces*, *Fisherfaces* e a técnica *SIFT* aplicada a imagens de rosto. Referente a estas técnicas podemos concluir que:

- A técnica *SIFT* é bastante mais pesada computacionalmente, tanto no que diz respeito a recursos espaciais como temporais. Neste caso, verificamos que os métodos de *Eigenfaces* e *Fisherfaces* necessitam de menos espaço de armazenamento da informação referente a uma imagem de rosto que a técnica *SIFT*. A comparação entre duas imagens de rosto, processadas com a técnica *SIFT*, é mais pesada pelo facto de uma imagem de rosto ser representada por vários descritores, onde cada descritor corresponde a um vector com dimensão igual a 128. Logo, para comparar duas imagens de rosto existe a necessidade de comparar os vários descritores das duas imagens de rosto enquanto que com os métodos de *Eigenfaces* e *Fisherfaces* apenas são comparados dois vectores.
- A técnica de *Eigenfaces* é a mais utilizada dos trabalhos que foram estudados nesta área, ainda que recentemente existam trabalhos com a aplicação da técnica *SIFT* a imagens de rosto.
- No trabalho que compara as três técnicas, a aplicação da técnica *SIFT* a imagens de rosto foi a que obteve os melhores resultados, seguida do método de *Fisherfaces* [37].

Em relação ao estudo das técnicas para a representação de imagens de rosto é de salientar que o objectivo inicial era implementar as 3 representações. No entanto, apenas foi utilizado o método de *Eigenfaces* devido aos seguintes factos:

1. Não foi encontrada nenhuma função de semelhança métrica que tivesse sido aplicada no método *SIFT* (ver capítulo 3);
2. O método de *Fisherfaces* requer um conjunto de imagens de treino para cada indivíduo da base de dados. Desta forma, o algoritmo de *Fisherfaces* exige um conhecimento prévio da base de dados em relação ao conjunto de imagens de rosto associado a cada indivíduo. Além disso, o método de *Fisherfaces* não pode ser aplicado a bases de dados de pequenas dimensões onde apenas existe 1 imagem de rosto associado a cada indivíduo [3].

### Capítulo 3. Medidas de semelhança entre imagens de rosto

Com base nas características extraídas de cada imagem de rosto, são usadas várias medidas de semelhança que permitem medir a similaridade entre os vários rostos. Neste caso, quanto menor for a medida entre duas imagens, maior será a similaridade entre elas. Entre as várias medidas de semelhança, usadas no âmbito do reconhecimento facial, existem algumas que são métricas.

Uma métrica ou distância é uma função  $d: E \times E \rightarrow \mathfrak{R}_0^+$ , onde  $E$  é um conjunto de elementos, que associa a cada par de elementos um número real não negativo que descreve o grau de similaridade entre os dois elementos e satisfaz as seguintes propriedades:

- Positividade:  $\forall x, y \in E \quad d(x, y) \geq 0 \wedge (d(x, y) = 0 \Leftrightarrow x = y)$  ;
- Reflexividade:  $\forall x \in E \quad d(x, x) = 0$  ;
- Simetria:  $\forall x, y \in E \quad d(x, y) = d(y, x)$  ;
- Desigualdade triangular:  $\forall x, y, z \in E \quad d(x, y) \leq d(x, z) + d(z, y)$  .

No nosso estudo, um elemento do conjunto  $E$  corresponde à representação de uma imagem de rosto.

De seguida serão apresentadas várias medidas de semelhança, que foram utilizadas em trabalhos na área de pesquisa de imagens de rosto. Para cada medida será apresentada a sua definição e serão apresentados alguns trabalhos relevantes que a utilizaram.

No anexo A2.1 é apresentada uma medida de semelhança, designada de *Earth Mover's Distance*, usada na área de reconhecimento de imagens. No entanto, não existem trabalhos relevantes na área de imagens de rosto que usem esta medida.

Nas duas últimas secções, deste capítulo, será apresentada uma pequena análise comparativa das funções estudadas com base nos trabalhos existentes, e uma conclusão com os contributos destas técnicas nesta dissertação.

### 3.1. Medidas de distância baseadas na métrica de *Minkowski*

As medidas de semelhança baseadas na métrica de *Minkowski*, também designadas como medidas da família LP [25], podem ser definidas como:

$$d(U, V) = \sqrt[p]{\sum_{i=1}^n |U_i - V_i|^p}$$

onde  $U$  e  $V$  são dois vectores de elementos de dimensão  $n$ ,  $U_i$  representa o elemento na posição  $i$  do vector  $U$ ,  $V_i$  representa o elemento na posição  $i$  no vector  $V$  e  $i \in [1, n]$ .

Existem três casos particulares da medida de *Minkowski* que dão origem às medidas de semelhança: Distância Euclidiana – L2, Distância de *Manhattan* – L1 e Distância de *Chebychev* –  $L^\infty$ , as quais serão descritas de seguida.

#### 3.1.1. Distância Euclidiana – L2

A distância Euclidiana é uma métrica que deriva da medida de *Minkowski* quando  $p = 2$ . Esta pode ser definida genericamente como:

$$d(U, V) = \sqrt{\sum_{i=1}^n (U_i - V_i)^2}$$

Existem vários trabalhos na área do reconhecimento facial que usam esta métrica. Um dos mais conhecidos corresponde ao método designado de *Eigenfaces* [2].

Tal como foi dito anteriormente, no método designado de *Eigenfaces* uma imagem de rosto é projectada no espaço de rostos como um vector de pesos. Seja  $n$  o número de vectores próprios usado, então cada imagem é representada por um vector de pesos  $X = \langle X_1, \dots, X_n \rangle$  com  $n$  elementos e a similaridade entre duas imagens é dada pela aplicação directa da distância Euclidiana. Quanto menor for a distância entre os respectivos vectores de duas imagens de rosto, maior será o grau de similaridade entre ambas.

Em [22] são apresentados três métodos para comparação de imagens de rosto com base nos descritores extraídos com a técnica *SIFT*. Os três métodos de comparação

diferem na representação e na forma como é feita o cálculo da função de semelhança. No primeiro, são comparados todos os descritores da primeira imagem com todos da segunda retornando o valor mínimo encontrado entre os vários pares de descritores, usando a distância Euclidiana. Em relação aos outros métodos, é feito o mesmo para as sub-imagens criadas e, no final, é feita a média das duas medidas. De seguida será apresentado, formalmente, o modo de comparação de duas imagens/sub-imagens que é usado nos três métodos.

Sejam  $S = \{s_1, \dots, s_m\}$  e  $T = \{t_1, \dots, t_n\}$  os conjuntos de descritores de imagens/sub-imagens pertencentes, respectivamente, a duas imagens  $T$  e  $S$ . A medida de semelhança entre elas é definida como:

$$D^{MPD}(S, T) = \min_{ij} ( d(s_i, t_j) )$$

A função  $d(s_i, t_j)$  representa a aplicação directa da distância Euclidiana entre dois descritores, onde cada descritor corresponde a um vector de 128 elementos. Assim,  $D^{MPD}(S, T)$  retorna a menor distância Euclidiana encontrada entre os vários pares de descritores. Apesar de usar na fórmula a distância Euclidiana, a  $D^{MPD}(S, T)$  não é uma métrica.

### 3.1.2. Distância de *Manhattan* – *L1*

A distância de *Manhattan*, é uma métrica que deriva da medida de *Minkowski* quando  $p = 1$  e é definida genericamente como:

$$d(U, V) = \sum_{i=1}^n |U_i - V_i|$$

Esta métrica pode ser também usada no método descrito anteriormente designado de *Eigenfaces* para comparar os respectivos vectores de pesos entre duas imagens de rosto.

Um trabalho onde podemos encontrar a referência à distância de *Manhattan* é o [5]. Neste, é feita uma avaliação de várias medidas em sistemas de reconhecimento facial baseados em *PCA – Principal Component Analysis* usando o método de *Eigenfaces*. As medidas de semelhança comparadas, além da distância de *Manhattan*, foram a distância

Euclidiana, Ângulo que corresponde à medida de semelhança Covariância (ver secção 3.2) definida no intervalo de  $[-1,1]$ , e a medida de *Yambor* referida como distância de *Mahalanobis* (ver secção 3.4.4). Neste trabalho foi também realizado um estudo sobre a combinação das várias medidas de semelhança, no entanto, não foram obtidas melhorias significativas na taxa de reconhecimento. No estudo em que as medidas de semelhança foram avaliadas individualmente, numa base de dados de imagens tiradas em momentos diferentes (no máximo com um ano de diferença), notou-se que a distância de *Manhattan* apresentou um desempenho ligeiramente superior à distância Euclidiana e Ângulo. Em relação ao teste com a base de dados de imagens tiradas no mesmo momento mas com diferentes expressões faciais, a distância de *Manhattan* obteve o melhor desempenho. No entanto, de uma forma geral, o melhor desempenho acabou por ser da medida de *Yambor*.

### 3.1.3. Distância de *Chebyshev* - $L_\infty$

A distância de *Chebyshev* -  $L_\infty$  deriva da medida de *Minkowski* quando  $p$  tende para mais infinito. Esta medida é métrica e é definida genericamente como:

$$d(U, V) = \text{Max}_i |U_i - V_i|$$

Como podemos verificar, esta medida corresponde ao maior dos módulos da diferença entre elementos de dois vectores com o mesmo índice.

No entanto, na pesquisa efectuada não foram encontrados trabalhos na área do reconhecimento facial que tivessem usado esta métrica.

## 3.2. Covariância

Esta medida de semelhança é baseada no co-seno do ângulo entre dois vectores. A fórmula original tem contradomínio definido no intervalo  $[-1,1]$ . Desta forma, esta foi modificada em [4] para ser definida no intervalo  $[0,2]$ , tal como podemos verificar de seguida:

$$d(U, V) = 1 - \frac{\sum_{i=1}^n U_i V_i}{\sqrt{\sum_{i=1}^n (U_i)^2} \sqrt{\sum_{i=1}^n (V_i)^2}}$$

onde  $U = \langle U_1, \dots, U_n \rangle$  e  $V = \langle V_1, \dots, V_n \rangle$  são dois vectores de elementos de dimensão  $n$ .

Podemos encontrar a referência a esta medida de semelhança no trabalho [4] *CSU – Face Identification Evaluation System* e em [5], na qual é denominada de Ângulo, que corresponde à formula original definida no intervalo  $[-1,1]$ . Neste último, a medida é avaliada num sistema de reconhecimento facial baseada em *PCA – Principal Component Analysis* usando vectores próprios. Desta forma, também pode ser usada no método de *Eigenfaces* com a aplicação directa da fórmula para medir a similaridade entre os vectores de pesos extraídos de cada imagem de rosto.

Esta medida de semelhança não é métrica [29].

### 3.3. Correlação

Esta medida de semelhança é baseada no coeficiente de correlação de Pearson.

A fórmula original tem contradomínio definido no intervalo  $[-1,1]$ . Desta forma, esta foi modificada em [4] para ser definida no intervalo  $[0,2]$ , tal como podemos verificar de seguida:

$$d(U, V) = 1 - \frac{\sum_{i=1}^n (U_i - \bar{U})(V_i - \bar{V})}{\sqrt{\sum_{i=1}^n (U_i - \bar{U})^2} \sqrt{\sum_{i=1}^n (V_i - \bar{V})^2}}$$

$$\bar{U} = \frac{\sum_{i=1}^n U_i}{n} \quad \bar{V} = \frac{\sum_{i=1}^n V_i}{n}$$

Neste caso,  $U = \langle U_1, \dots, U_n \rangle$  e  $V = \langle V_1, \dots, V_n \rangle$  são dois vectores de elementos de dimensão  $n$ .  $\bar{U}$  e  $\bar{V}$  representam respectivamente a média aritmética de todos os elementos do vector  $U$  e  $V$ .

Em [30] verificamos que não é uma métrica.

Esta medida de semelhança é referida também no *CSU – Face Identification*

*Evaluation System* [4].

### 3.4. Mahalanobis

De seguida serão apresentadas as medidas de semelhança *Mahalanobis – L1*, *Mahalanobis – L2*, *Mahalanobis – Co-seno*, e medida de *Yambor*.

Para a definição das quatro medidas deve ser considerado que  $U = \langle U_1, \dots, U_n \rangle$  e  $V = \langle V_1, \dots, V_n \rangle$  correspondem a dois vectores de pesos que descrevem duas imagens e  $\lambda = \langle \lambda_1, \dots, \lambda_n \rangle$  representa o vector de valores próprios. Em todas as fórmulas, os vectores têm uma dimensão  $n$  igual ao número de vectores próprios usado, onde  $U_i$ ,  $V_i$  e  $\lambda_i$  representam, respectivamente, o elemento na posição  $i$  dos vectores  $U$ ,  $V$  e  $\lambda$ .

#### 3.4.1. Mahalanobis – L1

A *Mahalanobis – L1* é uma métrica que corresponde à distância de *Manhattan – L1* projectada no espaço de *Mahalanobis* e é definida como:

$$d(U, V) = \sum_{i=1}^n \left| \frac{U_i - V_i}{\sqrt{\lambda_i}} \right|$$

Podemos encontrar o uso desta métrica no trabalho de investigação *CSU – Face Identification Evaluation System* [4] em que é apresentado um sistema para avaliação de desempenho de sistemas de reconhecimento facial.

#### 3.4.2. Mahalanobis – L2

A *Mahalanobis – L2* é uma métrica que corresponde à distância Euclidiana – *L2* projectada no espaço de *Mahalanobis* e é definida como:

$$d(U, V) = \sqrt{\sum_{i=1}^n \frac{(U_i - V_i)^2}{\lambda_i}}$$

Esta métrica é usada no trabalho de investigação *CSU – Face Identification Evaluation System* [4] com o método de *Eigenfaces*.

### 3.4.3. Mahalanobis – Co-seno

A *Mahalanobis – Co-seno* corresponde ao co-seno do ângulo entre dois vectores projectado no espaço de *Mahalanobis* e é definida como:

$$d(U, V) = - \frac{\sum_{i=1}^n \frac{U_i V_i}{\lambda_i}}{\sqrt{\sum_{i=1}^n \frac{(U_i)^2}{\lambda_i}} \sqrt{\sum_{i=1}^n \frac{(V_i)^2}{\lambda_i}}}$$

Em relação às outras três medidas (*Mahalanobis - L1*, *Mahalanobis - L2* e medida de *Yambor*) esta foi a que demonstrou melhores resultados no *CSU – Face Identification Evaluation System* [4] e por essa razão é usada como medida padrão para o sistema.

A *Mahalanobis – Co-seno* não é uma métrica.

### 3.4.4. Medida de *Yambor*

A medida de semelhança de *Yambor* é definida da seguinte forma:

$$d(U, V) = - \sum_{i=1}^n \frac{1}{\sqrt{\lambda_i}} U_i V_i$$

Esta é usada com o método de *Eigenfaces* baseado em *PCA* em [4], tal como as três medidas anteriores.

Outro trabalho onde podemos encontrar o uso desta medida de semelhança é o [5], onde é designada de distância de *Mahalanobis*. Neste trabalho é feito um estudo de várias medidas de semelhança incluindo a combinação destas em sistemas de reconhecimento facial baseados em *PCA* usando o método de *Eigenfaces*. As outras

medidas de semelhança comparadas foram a distância de *Manhattan*, distância Euclidiana e Ângulo. O resultado final em [5] demonstrou que a medida de semelhança de *Yambor* foi a que obteve melhor desempenho, quando comparada com as outras três. A combinação das várias medidas não resultou num ganho significativo de desempenho.

A medida de semelhança de *Yambor* não é uma métrica.

### 3.5. Distância de *Hausdorff*

De uma forma genérica, a distância de *Hausdorff* entre dois conjuntos de pontos corresponde à distância máxima de um conjunto em relação ao elemento mais próximo do outro conjunto [8]. Assim, a distância de *Hausdorff* entre dois conjuntos de pontos  $A$  e  $B$  é definida como:

$$H(A, B) = \max(h(A, B), h(B, A))$$

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

A medida  $h(A, B)$  pode ser calculada comparando a distância de cada elemento do conjunto  $A$  com todos os elementos do conjunto  $B$ , de modo a determinar, para cada elemento de  $A$ , a distância deste ao elemento mais próximo de  $B$ . De seguida, deste conjunto de valores é retornada a maior distância. Como a distância  $h(A, B)$  pode ser diferente da  $h(B, A)$ , a distância de *Hausdorff* é definida como o máximo destas duas distâncias.

Na fórmula apresentada,  $\|a - b\|$  representa a norma euclidiana ou distância Euclidiana entre dois pontos  $a$  e  $b$ .

Considerando  $A$  e  $B$  como conjuntos de pontos característicos de dois objectos, a distância de *Hausdorff* pode ser usada como uma medida para calcular a similaridade entre dois objectos. Nesta definição genérica da distância de *Hausdorff* são comparados pontos. Esta é uma métrica, tal como podemos verificar em [18].

### 3.5.1. Medida parcial de *Hausdorff*

A variante da distância de *Hausdorff* definida como medida parcial de *Hausdorff* [8] e é dada pela seguinte fórmula:

$$Hp(A, B) = \max(hp(A, B), hp(B, A))$$

$$hp(A, B) = Kth \min_{a \in A} \min_{b \in B} \|a - b\|$$

A medida  $hp(A, B)$  pode ser calculada comparando a distância de cada elemento do conjunto  $A$  com todos os elementos do conjunto  $B$ , de modo a determinar, para cada elemento de  $A$ , a distância deste ao elemento mais próximo de  $B$ . A única diferença em relação à medida original é que deste conjunto de valores é retornada a  $k$ -ésima maior distância em vez da maior. Na fórmula apresentada  $\|a - b\|$  representa a norma euclidiana entre dois pontos  $a$  e  $b$ . Para  $k = 1$ , a fórmula transforma-se na primeira medida apresentada como distância de *Hausdorff*.

A medida parcial de *Hausdorff* não é uma métrica [18].

#### 3.5.1.1. Medida de *Hausdorff* dos níveis de cinzento

A medida de *Hausdorff* dos níveis de cinzento (termo em inglês, *Gray Hausdorff Distance* [8]) que de seguida se apresenta é baseada nas duas medidas de semelhança descritas anteriormente.

Sejam  $M$  e  $T$  duas colecções de elementos em que cada elemento  $M_i$  e  $T_i$  contém um conjunto de pontos. A medida de *Hausdorff* dos níveis de cinzento entre  $M$  e  $T$  é definida como:

$$Hg(M, T) = \max(hpg(M, T), hpg(T, M))$$

$$hpg(M, T) = Kth_{\substack{i=0, 2^n-1 \\ m \in M_i}} d(m, T_i)$$

$$d(m, T_i) = \begin{cases} \min_{t \in T_i} \|m - t\|, & \text{se o conjunto } T \text{ não for vazio} \\ L, & \text{se o conjunto for vazio} \end{cases}$$

Na fórmula apresentada  $\|m - t\|$  representa a norma euclidiana entre os pontos  $m$  e  $t$ , ou seja, a distância do ponto  $m$  ao ponto  $t$ , e  $L$  é uma constante. Neste caso,  $hpg(M, T)$  é definida como a medida directa entre  $M$  e  $T$  e pode ser calculada da seguinte forma: Cada ponto  $m$  de  $M_i$  é comparado com o conjunto de pontos  $T_i$ , ou seja, são comparados sempre pontos relacionados pelo mesmo índice  $i$ . O objectivo consiste em determinar a distância de cada ponto  $m$  de  $M_i$  ao ponto mais próximo de  $T_i$ . Neste caso, a cada ponto de  $M_i$  fica associado a distância ao vizinho mais próximo de  $T_i$ , ou seja, ao ponto mais próximo do conjunto  $T_i$ . Depois de calcular esta distância para todos os conjuntos é retornada a  $k$ -ésima maior distância. A medida de *Hausdorff* dos níveis de cinzento não é uma métrica.

Esta medida de semelhança é apresentada em [8] e permite medir a similaridade entre duas imagens de rosto em tons de cinzento. Neste caso, as duas imagens podem ser comparadas directamente, ou seja, não é necessário um conjunto de imagens de rosto de treino. A ideia fundamental deste método consiste em medir a distribuição de valores de cinzento para comparar a aparência dos rostos.

Para tal, são considerados conjuntos de pontos associados a cada valor possível para o *pixel*. Cada elemento deste conjunto é um ponto (uma coordenada) da imagem que contém o valor do *pixel* associado ao conjunto. Como são usados 8 *bits* ( $n = 8$ ) para a representação de cada *pixel* cinzento existirão no máximo  $2^n$  conjuntos de pontos ou seja 256 para cada imagem. No entanto, os *pixels* podem ser representados pelos  $n$  *bits* mais significativos, onde  $n < 8$ . Neste caso, o número total de conjuntos de pontos será menor. Uma vantagem de representar cada *pixel* pelos  $n$  *bits* mais significativos é que o sistema torna-se mais robusto a pequenas variações das condições de iluminação das imagens.

Desta forma, para medir a similaridade entre duas imagens são criadas duas novas  $QM$  e  $QT$  em que cada *pixel* é representado pelos  $n$  *bits* mais significativos. Com base nestas duas novas representações da imagem são criadas respectivamente as colecções  $M$  e  $T$ , em que cada uma contém  $2^n$  elementos. Neste caso, temos um elemento  $M_i$  e um elemento  $T_i$  que possuem o conjunto de todos os pontos, da primeira e da segunda imagem respectivamente, que têm a cor na escala de cinzento  $i$ .

Assim, a medida de *Hausdorff* dos níveis de cinzento entre duas imagens representadas por  $M$  e  $T$  é dado pela aplicação directa da fórmula  $Hg(M, T)$ . A constante  $L$  desta fórmula é definida como  $\sqrt{r^2 + c^2} + 1$  em que  $r$  e  $c$  representam as dimensões da imagem original.

Em [8] é apresentada uma forma eficiente de calcular esta medida de semelhança.

### 3.6. Análise comparativa

Esta secção apresenta os resultados de alguns trabalhos que comparam algumas das métricas que foram descritas.

Em [37] é realizada uma comparação entre um método que utilizou a técnica *SIFT* para a representação das imagens de rosto e os métodos designados de *Eigenfaces* e *Fisherfaces*. Nestes dois últimos, foram utilizadas como métricas a distância Euclidiana e a Distância de *Manhattan* e os resultados deste trabalho mostraram que, na maioria dos testes realizados, os dois métodos apresentaram melhores resultados com a distância de *Manhattan*.

Em [5] é apresentado um trabalho onde foi realizada uma avaliação de várias medidas de semelhança e que utiliza o método de *Eigenfaces* para a representação de imagens de rosto. Neste trabalho, os resultados dos testes que comparam a distância Euclidiana e a distância de *Manhattan* mostraram que a segunda obteve um desempenho superior.

Não foram encontrados trabalhos que comparem conjuntamente as métricas *Mahalanobis – L1*, *Mahalanobi2 – L2*, distância Euclidiana e distância de *Manhattan*.

Em relação à distância de *Hausdorff*, também não foi encontrado nenhum trabalho na área de reconhecimento facial onde esta tivesse sido comparada com outras métricas.

### 3.7. Conclusão e contributos

Neste capítulo foram descritas várias medidas de semelhança, usadas no âmbito de reconhecimento facial, que permitem medir a similaridade entre duas imagens de rosto, tendo por base as características extraídas destas.

Do conjunto de medidas de semelhança apresentado as 3 medidas da família LP, a *Mahalanobis – L1*, a *Mahalanobis – L2*, e a *Hausdorff* são métricas.

Não foram encontradas funções de semelhança métricas que permitam medir a similaridade entre duas imagens de rosto representadas por descritores extraídos com a técnica *SIFT*.

Uma conclusão relevante que podemos retirar do capítulo 2 e 3 é que é possível definir espaços métricos no domínio das imagens de rosto.

Dos espaços métricos analisados, 4 foram utilizados nesta dissertação, utilizando o método de *Eigenfaces* para a representação de imagens de rosto. As métricas usadas são as distâncias Euclidiana, *Manhattan*, *Mahalanobis – L1* e *Mahalanobis – L2*. Não foi encontrado nenhum trabalho que compare estas 4 métricas. No entanto, em [37] os métodos designados de *Eigenfaces* e *Fisherfaces* foram comparados com a distância Euclidiana e de *Manhattan*. Nesta experiência, os dois métodos obtiveram, na maioria dos testes realizados, uma proporção de resultados verdadeiros superior quando utilizaram a distância de *Manhattan*.

## Capítulo 4. Pesquisa por semelhança em espaços métricos

A ideia fundamental da pesquisa por semelhança, ou também denominada pesquisa aproximada, consiste em obter o conjunto de elementos da base de dados similares a um determinado elemento de consulta. A semelhança entre dois elementos é dada com base numa medida de semelhança.

A pesquisa numa base de dados, pelos elementos semelhantes, pode conduzir a uma procura exaustiva na medida em que é necessário calcular a medida de semelhança entre o elemento da consulta e todos os elementos da base de dados. De modo a evitar a procura exaustiva de elementos semelhantes, é necessário organizar os elementos da base de dados em zonas/regiões, de tal modo que elementos semelhantes se encontrem na mesma zona, para que a pesquisa seja realizada só em algumas zonas. Um conceito matemático muito utilizado quando é necessário medir a proximidade/semelhança de elementos é o espaço métrico.

Muitas estruturas de dados foram propostas, baseadas no conceito de espaço métrico, que têm o objectivo de tornar as pesquisas por semelhança mais eficientes. Estas procuram minimizar o número de cálculos da medida de semelhança aquando de uma pesquisa. Para tal, particionam o espaço em zonas com base na medida de semelhança, a qual mede a semelhança/proximidade entre elementos. Estas estruturas de dados denominam-se estruturas de dados métricas.

Para que seja possível o uso destas estruturas, é necessário que a medida de semelhança, que mede a similaridade entre os vários elementos da base de dados, seja métrica, de modo a poder definir um espaço métrico.

Um espaço métrico corresponde a um par  $(D, d)$  onde  $D$  corresponde ao conjunto de elementos denominado universo e  $d: D \times D \rightarrow \mathfrak{R}_0^+$ , corresponde a uma medida de distância métrica.

Dependendo do tipo de organização dos elementos da base de dados, podemos classificar as estruturas de dados métricas como baseadas em *pivots* ou em

agrupamentos (termo em inglês, *clusters*). Apesar das muitas diferenças entre as várias estruturas de dados métricas, todas se baseiam, fundamentalmente, nas propriedades da função de distância, nomeadamente a desigualdade triangular e simetria, de modo a minimizar os cálculos da função de distância. Ou seja, numa pesquisa, vários elementos podem logo ser descartados sem que seja calculada a distância em relação ao elemento de consulta.

Em relação às pesquisas por similaridade, existem dois tipos: pesquisas por alcance (termo em inglês, *range queries*) e pesquisas dos  $k$  vizinhos mais próximos (termo em inglês, *k-nearest neighbor queries*). Nesta dissertação só estamos interessados na pesquisa por alcance, ou seja, em encontrar o conjunto de elementos da base de dados cuja distância em relação ao elemento de consulta não excede um determinado raio de pesquisa. Neste caso, o raio de pesquisa corresponde a um valor real não negativo. Mais formalmente, uma pesquisa por alcance  $(q,r)_d$  é definida pelo conjunto  $\{u \in U / d(q,u) \leq r\}$  onde  $U$  corresponde ao conjunto de elementos da base de dados,  $d$  é a função de distância que mede a similaridade entre dois elementos,  $q$  é o elemento de pesquisa e  $r$  é o raio de pesquisa.

Este capítulo está organizado em 4 secções. Na secção 1, são descritos os dois tipos de estruturas de dados métricas baseadas em *pivots* ou em agrupamentos. Na secção 2 é realizado um estudo de várias estruturas de dados métricas propostas com o objectivo de tornar as pesquisas por alcance mais eficientes. Na secção 3 é apresentada uma análise comparativa das estruturas de dados métricas, no que diz respeito à sua aplicabilidade e eficiência em vários domínios de aplicação. Na secção 4 são apresentadas as conclusões e os contributos.

## 4.1. Classificação das estruturas de dados métricas

As estruturas de dados métricas podem ser classificadas como baseadas em *pivots* ou baseadas em agrupamentos [19], dependendo do método utilizado para particionar os elementos da base de dados.

Em relação às estruturas baseadas em *pivots* verificamos que vários elementos da base de dados são seleccionados como *pivots*, e todos os outros elementos são classificados de acordo com a distância em relação a estes. Desta forma, nos métodos

baseados em *pivots* consideramos que dois elementos  $x$  e  $y$  estão na mesma zona se estiverem à mesma distância em relação a todos os *pivots* definidos. Por exemplo, na figura 4.1 retirada de [19], o espaço foi particionado com base em 2 *pivots*  $u_8$  e  $u_{11}$ . Desta forma, verificamos que do ponto de vista do *pivot*  $u_8$ , os elementos  $u_2$  e  $u_4$  estão próximos. Assim, a pesquisa dos elementos próximos de um dado elemento  $q$  consiste em procurar os elementos nas intersecções de algumas das esferas centradas nos *pivots*. No exemplo da figura 4.1, os candidatos ao resultado da pesquisa com um elemento de consulta  $q$  e raio  $r$  seriam  $u_5$  e  $u_{12}$ . Neste caso, verificamos que  $u_5$  não está próximo da zona de pesquisa e isto deve-se ao facto da partição baseada em 2 *pivots* não garantir a proximidade dos elementos da mesma zona.

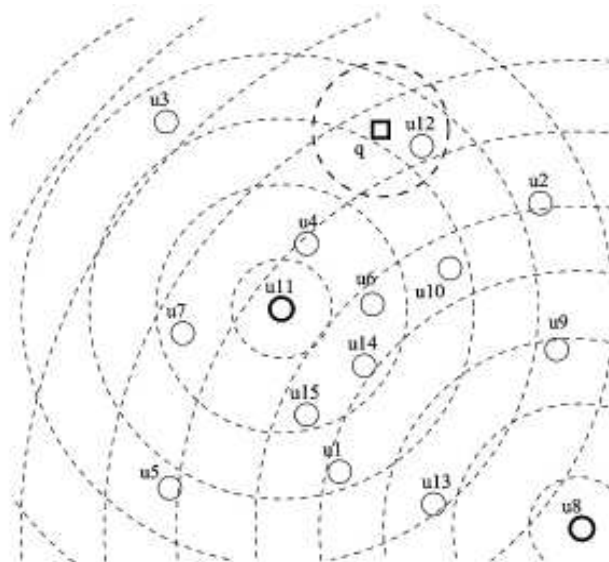


Figura 4.1 – Partição do espaço com base em 2 *pivots*

De modo a procurar garantir a proximidade dos elementos na mesma zona, é normal utilizar um maior número de *pivots*. Desta forma, verificamos que os algoritmos baseados em *pivots* geralmente têm a tendência a melhorar a eficiência das pesquisas à medida em que são usados mais *pivots*. No entanto, é necessário ter em conta que quantos mais *pivots* forem usados, maior será a quantidade de memória utilizada.

A propriedade da desigualdade triangular pode ser usada aquando de uma pesquisa por alcance  $(q, r_q)_d$  para descartar vários elementos da base de dados, utilizando a informação dos *pivots*, sem calcular a distância em relação a esses elementos. Por exemplo, se um *pivot*  $p_1$  guardar a distância  $d(p_1, x)$  em relação a um elemento  $x$ , verificamos que um elemento  $x$  pode ser descartado da pesquisa se  $|d(q, p_1) - d(x, p_1)| >$

$r_q$  como podemos verificar pela figura 4.2(a). Neste caso, verificamos que o ponto  $x$  não está contido na zona de pesquisa. No entanto, existem casos em que um *pivot* não nos dá informação suficiente para descartar o elemento da pesquisa, como podemos verificar na figura 4.2(b). Na figura 4.2(c) verificamos que o elemento  $x$  já poderia ser descartado, pois através de  $p2$  verificamos que  $|d(q,p2) - d(x,p2)| > r_q$ . Desta forma, generalizando verificamos que um elemento pode ser descartado da pesquisa se existir um *pivot*  $p$  tal que  $|d(q,p) - d(x,p)| > r_q$  pois, neste caso, pela propriedade da desigualdade triangular a distância  $d(q,x) > r_q$ . A razão pela qual podemos descartar o elemento  $x$  será demonstrada de seguida:

1. Pela propriedade da desigualdade triangular sabemos que  $d(p,x) \leq d(p,q) + d(q,x)$ ;
2. Pela propriedade da desigualdade triangular sabemos que  $d(q,p) \leq d(q,x) + d(x,p)$ ;
3. De (1) sabemos que  $d(p,x) - d(p,q) \leq d(q,x)$ ;
4. De (2) sabemos que  $d(q,p) - d(x,p) \leq d(q,x)$ ;
5. Pela propriedade da simetria e por (3) temos que  $d(x,p) - d(q,p) \leq d(q,x)$ ;
6. Por (4) e (5) temos que  $|d(q,p) - d(x,p)| \leq d(q,x)$ .

Logo se  $|d(q,p) - d(x,p)| > r_q$  então pela propriedade da transitividade  $d(q,x) > r_q$  e, desta forma, o elemento  $x$  pode ser descartado.

A figura 4.2(d) mostra um exemplo onde podemos seleccionar um elemento  $x$  sem calcular a distância  $d(q,x)$ . Neste caso verificamos que se  $d(q,p1) + d(p1,x) \leq r_q$ , pela propriedade da desigualdade triangular garantidamente  $d(q,x) \leq r_q$ .

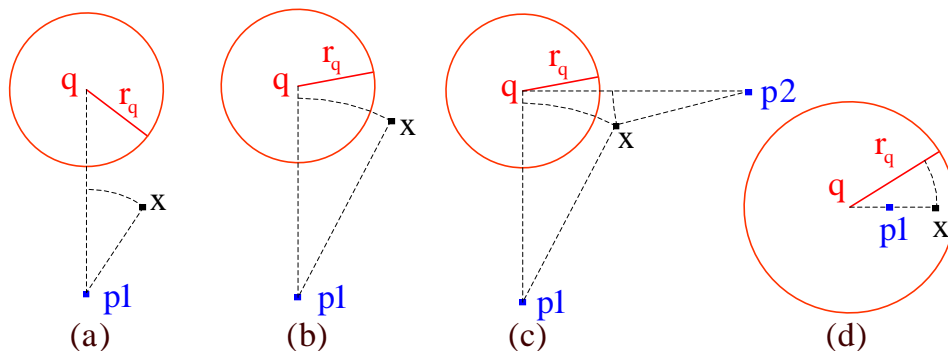


Figura 4.2 – Descarte e selecção de elementos com *pivots*

A *VPTree – Vantage Point Tree* e a *LAESA – Linear Approximating Eliminating Search Algorithm* são dois exemplos de estruturas de dados métricas baseadas em *pivots*.

No que diz respeito às estruturas baseadas em agrupamentos, notamos que o conjunto de elementos da base de dados é dividido em várias zonas o mais compactas possíveis denominadas agrupamentos. Desta forma, depois de seleccionados os centros dos agrupamentos os vários elementos são associados ao centro mais próximo. Neste caso, verificamos que o centro e o *pivot* desempenham um papel muito semelhante, no entanto, a diferença reside em que um dado elemento  $x$  está associado a um *pivot*  $p$  com base na distância entre  $p$  e  $x$  e não porque  $p$  seja o *pivot* mais próximo de  $x$ .

Na figura 4.3 retirada de [19] verificamos, que o espaço é particionado em 4 agrupamentos com centros  $c1, c2, c3$  e  $c4$ . Neste caso, é ilustrado um espaço de pesquisa representado a tracejado definido pelo elemento  $q1$  e por um determinado raio  $r$ , onde verificamos que apenas são pesquisados os agrupamentos que intersectam este espaço ou seja  $c1, c2$  e  $c3$ . Neste caso, o resultado desta pesquisa seria os elementos  $u11$  e  $u4$ .

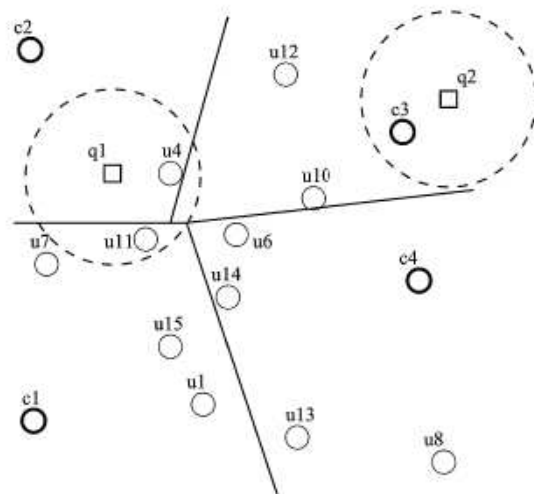


Figura 4.3 – Partição do espaço em 4 agrupamentos

A propriedade da desigualdade triangular pode ser usada aquando de uma pesquisa por alcance  $(q, r_q)_d$  para descartar vários elementos que pertençam a um determinado agrupamento como podemos verificar pela figura 4.4(a).

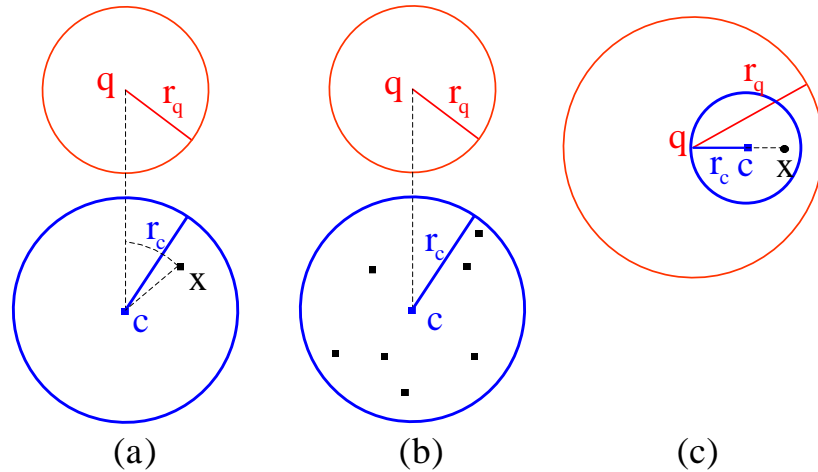


Figura 4.4 – Descarte e selecção de elementos num agrupamento

Na figura 4.4(a) verificamos que se um elemento  $x$  pertence a um agrupamento  $c$  e se  $|d(q,c)-d(c,x)| > r_q$  então, pela propriedade da desigualdade triangular,  $d(q,x) > r_q$  e  $x$  pode ser descartado da pesquisa sem que  $d(q,x)$  seja calculada. Da mesma forma, como sabemos que todos os elementos do agrupamento estão à distância máxima  $r_c$  em relação ao seu centro  $c$  e se  $|d(q,c)-r_c| > r_q$  então garantidamente todos os elementos do agrupamento podem ser descartados, pois como podemos verificar na figura 4.4(b) não fazem parte do espaço da pesquisa. A figura 4.4(c) mostra um exemplo de como vários elementos pertencentes a um agrupamento  $c$  podem ser seleccionados sem calcular a medida de distância em relação aos elementos. Pela propriedade da desigualdade triangular verificamos que se  $d(q,c)+d(c,x) \leq r_q$  então  $d(q,x) \leq r_q$ . Como todos os elementos do agrupamento estão a uma distância máxima  $r_c$  do centro  $c$ , se  $d(q,c)+r_c \leq r_q$  então todos os elementos fazem parte do conjunto de resultados.

A *LC – List of Clusters*, *RLC – Recursive Lists of Clusters*, *GNAT – Geometric Near-neighbor Access Tree* e *DSAT – Dynamic Spatial Approximation Tree* são alguns exemplos de estruturas de dados métricas baseadas em agrupamentos.

## 4.2. Estruturas de dados métricas

Nesta secção serão apresentadas as várias estruturas de dados métricas que foram utilizadas nesta dissertação.

### 4.2.1. LC – List of Clusters

A estrutura de dados métrica estática denominada de *LC – List of Clusters* [23], tal como o próprio nome indica, organiza os elementos numa sequência de agrupamentos. Cada agrupamento corresponde a um triplo  $(c,r,I)$  que contém um elemento  $c$  pertencente à base de dados  $B$  que representa o centro do agrupamento, um número real não negativo  $r \in \mathbb{R}_0^+$  que corresponde ao raio do agrupamento, e um subconjunto  $I$  de elementos da base de dados em que a distância em relação ao centro do agrupamento não excede o raio  $r$ . O centro do agrupamento não faz parte do subconjunto de elementos  $I$  e por esta razão a distância dos vários elementos ao centro é sempre maior que 0 logo,  $I = \{x \in B / 0 < d(x,c) \leq r\}$ . Formalmente, uma *LC* referente à base de dados  $B$  tem a forma  $L = \langle (c_1, r_1, I_1), \dots, (c_n, r_n, I_n) \rangle$ , onde  $n$  corresponde ao número agrupamentos e  $\bigcup_{i=1..n} I_i \cup \{c_i\} = B$ . A figura 4.5 descreve uma *LC* onde  $L = \langle (c_1, r_1, \{x_1, x_2, x_3\}), (c_2, r_2, \{x_4, x_5\}), (c_3, r_3, \{x_6, x_7\}) \rangle$ .

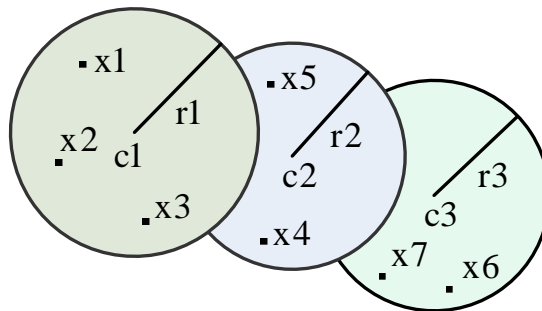


Figura 4.5 – Partição do espaço em agrupamentos na *LC*

Os agrupamentos da lista são disjuntos, ou seja, agrupamentos diferentes não têm elementos em comum. Isto deve-se ao facto de cada elemento apenas ocorrer no primeiro agrupamento no qual se pode encaixar (de acordo com as restrições descritas anteriormente). Isto significa que se um elemento ocorrer num agrupamento, a distância do elemento a todos os centros dos agrupamentos anteriores da lista é maior que o raio dos agrupamentos. Por exemplo, na figura 4.5 o elemento  $x_6$  que se encontra no agrupamento com centro  $c_3$  garantidamente verifica  $d(x_6, c_1) > r_1$  e  $d(x_6, c_2) > r_2$ .

### 4.2.1.1. Algoritmo de construção

Na figura 4.6 é apresentada uma breve descrição dos vários passos do algoritmo de construção da *LC*, dada uma base de dados *B* de elementos e sendo ‘:’ o operador de construção de listas.

Construção-LC( <i>B</i> ) Se $B = \emptyset$ Retornar a lista vazia; Seleccionar um elemento $c \in B$ para ser o centro do novo agrupamento; Seleccionar um raio $r_c$ para o agrupamento; $I \leftarrow \{b \in B - \{c\}, d(c,b) \leq r_c\}$ ; $E \leftarrow B - I$ ; Retornar $(c, r_c, I)$ : Construção-LC( <i>E</i> );
---

Figura 4.6 – Construção da *LC*

Neste algoritmo existem 2 aspectos importantes a ter em conta que correspondem à selecção do centro e à selecção do raio de cada agrupamento.

Em relação à selecção do centro, os autores, em [23], apresentam 5 alternativas:

1. Um elemento aleatório;
2. O elemento mais próximo em relação ao centro do agrupamento anterior;
3. O elemento mais distante em relação ao centro do agrupamento anterior;
4. O elemento que minimiza as distâncias em relação aos centros escolhidos anteriormente;
5. O elemento que maximiza as distâncias em relação aos centros escolhidos anteriormente.

No que diz respeito à selecção do raio, em [23], verificamos que existem duas opções: raio de tamanho fixo ou definido de modo a que os agrupamentos tenham um número fixo de elementos. Nos agrupamentos com raio fixo verificamos que estes têm tendência a ficar vazios à medida que avançamos na lista. Na segunda alternativa verificamos que à medida que avançamos na lista os agrupamentos têm tendência para englobar áreas cada vez maiores.

### 4.2.1.2. Algoritmo de pesquisa por alcance

O algoritmo de pesquisa por alcance está descrito na figura 4.7. Este recebe como parâmetros a lista de agrupamentos *L*, o elemento de pesquisa *q* e o raio  $r_q$  da pesquisa.

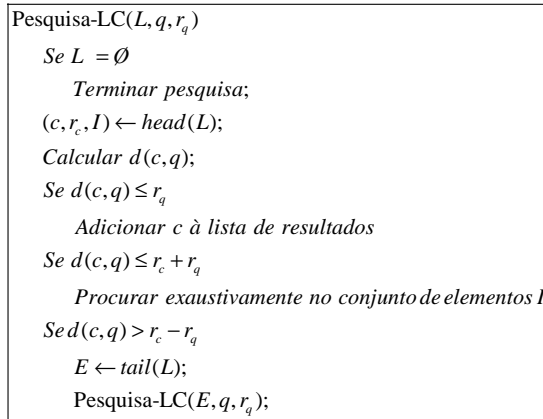


Figura 4.7 – Pesquisa por alcance na LC

Neste algoritmo a lista de agrupamentos é percorrida. Para cada agrupamento é calculada a distância do centro do agrupamento ao elemento de pesquisa. Se o centro  $c$  pertencer à pesquisa (condição  $d(c, q) \leq r_q$ ) então este é adicionado ao conjunto de resultados. De seguida, o conjunto de elementos no agrupamento é pesquisado exaustivamente se o espaço que a pesquisa abrange intersectar o agrupamento (condição  $d(c, q) \leq r_c + r_q$ ), como podemos verificar na figura 4.8.

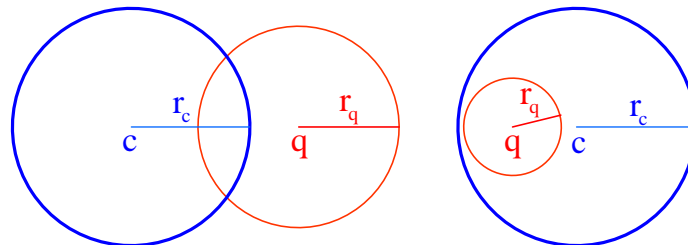


Figura 4.8 – Duas situações em que o espaço de pesquisa intersecta o agrupamento

Para além disso, também é verificado se o espaço de pesquisa é disjunto do espaço do agrupamento (condição  $d(c, q) > r_c + r_q$ ), de modo a descartar todos os elementos do agrupamento sem calcular as distâncias, como podemos verificar quando ocorre a situação descrita na figura 4.9(a).

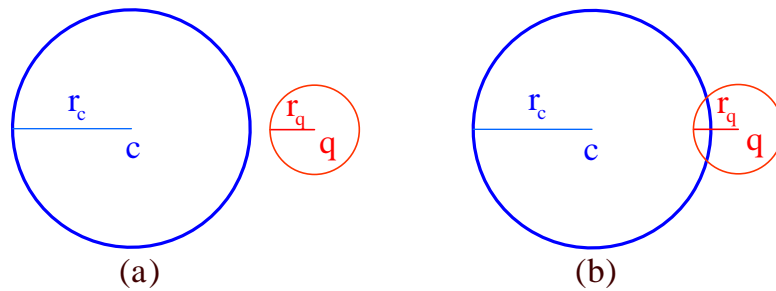


Figura 4.9 – Duas situações onde é necessário pesquisar nos agrupamentos seguintes

A pesquisa prossegue no agrupamento seguinte, se o espaço de pesquisa não estiver totalmente contido dentro do agrupamento (condição  $d(c, q) > r_c - r_q$ ) como podemos verificar nas situações apresentadas na figura 4.9.

#### 4.2.2. *RLC – Recursive Lists Of Clusters*

A *RLC – Recursive Lists Of Clusters* apresentada em [41, 6] é uma estrutura de dados métrica que se baseou na *LC*.

O tamanho da lista de agrupamentos afecta naturalmente a eficiência das várias operações que podem ser realizadas sobre a estrutura. Por outro lado, o aumento do raio de alcance de cada agrupamento conduz a listas de menor tamanho, no entanto, neste caso verificamos que os primeiros agrupamentos da lista ficam bastante carregados em termos do número de elementos. Desta forma, na *RLC* é proposta uma adaptação na estrutura anterior de modo a compensar este problema.

A maior diferença que encontramos na *RLC* em relação à *LC* reside na forma de organização dos elementos dentro de cada agrupamento. Na *RLC*, a lista de elementos *I* de cada agrupamento é implementada num vector se o número de elementos não ultrapassar uma determinada capacidade, caso contrário o interior do agrupamento é implementado como uma lista de agrupamentos (ou seja, a própria estrutura). A figura 4.10 retirada de [6] apresenta uma *RLC* em que a capacidade de cada vector tem uma de 5 elementos.

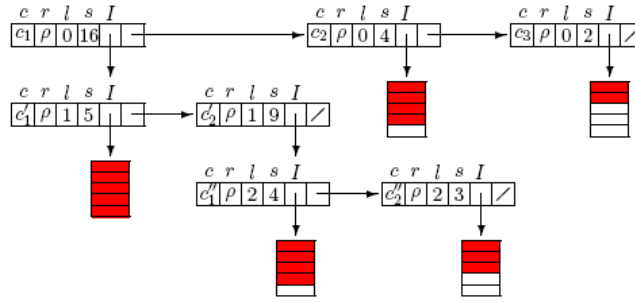


Figura 4.10 – Exemplo de uma *RLC*

Na *RLC* descrita em [6] são considerados agrupamentos com um raio fixo de modo a lidar com os problemas resultantes de bases de dados dinâmicas. Nesta estrutura de dados não se optou por agrupamentos com um tamanho fixo de elementos, já que têm a desvantagem de levar à impossibilidade de realizar actualizações de forma eficiente.

A *RLC* é uma estrutura de dados métrica dinâmica que, de modo a tornar as pesquisas mais eficientes, utiliza várias estratégias na organização dos elementos dentro dos agrupamentos e mantém para cada elemento uma sequência de distâncias, as quais dizem respeito às distâncias entre o elemento e os centros dos agrupamentos a que este pertence. Esta sequência encontra-se ordenada de acordo com a relação pai-filho aos centros dos agrupamentos. Para além disso, os elementos guardados no vector estão ordenados de forma decrescente de acordo com a distância ao centro do agrupamento a que pertencem.

#### 4.2.2.1. Algoritmo de construção

A construção da *RLC* é feita por uma sucessão de operações de inserção de elementos. A forma da estrutura de dados depende da ordem de inserção dos elementos.

#### 4.2.2.2. Algoritmo de inserção

A operação de inserção de um novo elemento  $x$  na *RLC* pode ser realizada percorrendo a lista de agrupamentos (por exemplo  $\langle (c_1, r, I_1), \dots, (c_n, r, I_n) \rangle$ ) até que seja encontrado um agrupamento que deva conter o elemento, isto é, a distância do elemento ao centro não excede o raio do agrupamento (condição  $d(x, c_i) \leq r$ ). Podem existir vários agrupamentos que podem conter o elemento, no entanto, este é inserido no

primeiro agrupamento encontrado, iterando a lista de agrupamentos.

No caso de não existir nenhum agrupamento que possa conter o elemento, um novo agrupamento é criado no final da lista sendo o elemento inserido como o centro do agrupamento (ficando no exemplo,  $\langle (c_1, r, I_1), \dots, (c_n, r, I_n), (x, r, \emptyset) \rangle$ ). A descrição dos vários passos do algoritmo recursivo para inserir um elemento na lista está na figura 4.11.

```

Inserir-Elemento-RLC(L, el)
  Se L = ∅
    Criar novo agrupamento C ← (el, r, ∅);
    Retornar lista C : L;
  (c, r, I) ← head(L);
  T ← tail(L);
  Se d(c, el) ≤ r
    Se c = el
      Retornar L;
    Inserir dentro do agrupamento C ← (c, r, {I ∪ el});
    Retornar lista C : T;
  Retornar lista (c, r, I) : Inserir-Elemento-RLC(T, el);

```

Figura 4.11 – Inserção de um novo elemento na RLC

Um aspecto importante está relacionado com a inserção do elemento dentro do agrupamento  $(c_i, r, I_i)$ , ou seja, no subconjunto de elementos  $I_i$ . Se este for representado por um vector, e se existir espaço para um elemento adicional então este é inserido no vector. O elemento é inserido ordenadamente no vector de forma decrescente em relação à distância ao centro do agrupamento. No caso do elemento já existir no vector, nada é feito. Quando não existe espaço no vector é criada uma nova lista de agrupamentos e os vários elementos do vector são adicionados nesta lista por sucessivas operações de inserção. Neste caso, o vector é substituído por uma lista de agrupamentos.

Por outro lado, no caso em que os elementos dentro do agrupamento são representados por uma lista recursiva de agrupamentos, a inserção do elemento é aplicada recursivamente nessa lista (como uma operação normal de inserção na RLC).

### 4.2.2.3. Algoritmo de remoção

Para realizar a operação de remoção de um elemento  $x$  é necessário percorrer os vários agrupamentos (por exemplo  $\langle (c_1, r, I_1), \dots, (c_n, r, I_n) \rangle$ ), e em cada iteração  $i$ , é

calculada a distância do elemento a remover ao centro do agrupamento  $d(c_i, x)$ . Se esta distância for menor que o raio deste então o agrupamento poderá conter o elemento a remover. Se não for encontrado nenhum agrupamento em que esta última condição se verifique então o elemento não existe na lista e o algoritmo termina. Em relação à remoção do elemento no agrupamento, existe o caso especial em que o elemento a remover corresponde ao centro do agrupamento ( $c_i = x$ ). Neste caso, o procedimento consiste em remover o agrupamento e reinserir os elementos de  $I_i$  na lista. Caso contrário é feita a remoção dentro do agrupamento.

A descrição dos vários passos do algoritmo de remoção de um elemento na lista está na figura 4.12.

```

Remover-Elemento-RLC( $L, el$ )
  Se  $L = \emptyset$ 
    Retornar  $L$ ;
   $(c, r, I) \leftarrow head(L)$ ;
   $T \leftarrow tail(L)$ ;
  Se  $d(c, el) \leq r$ 
    Se  $c = el$ 
      Reinserir cada elemento de  $I$  em  $T$ ;
      Retornar  $T$ ;
    Remover dentro do agrupamento  $C \leftarrow (c, r, \{I\} - el)$ ;
    Retornar lista  $C : T$ ;
  Retornar lista  $(c, r, I) : Remover-Elemento-RLC(T, el)$ ;

```

Figura 4.12 – Remoção de um elemento na RLC

Em relação à remoção dentro do agrupamento, se o subconjunto de elementos for representado pelo vector o elemento é removido deste. Caso contrário, a operação de remoção é aplicada recursivamente à lista de agrupamentos. Neste caso, sempre que o número de elementos é igual à capacidade de um vector, o subconjunto de elementos é transformado num vector.

#### 4.2.2.4. Algoritmo de pesquisa por alcance

A organização dos elementos do agrupamento e a manutenção em cada elemento da sequência de distâncias deste aos centros dos agrupamentos a que pertence, permitem otimizar a operação de pesquisa por alcance, procurando minimizar o número de cálculos da medida de distância entre os vários elementos.

A operação de pesquisa por alcance  $(q, r_q)_d$  necessita de procurar todos os elementos na *RLC* cuja distância ao elemento  $q$  não exceda o valor  $r_q$ . Logo, o algoritmo proposto nesta estrutura de dados percorre toda a lista recursiva de agrupamentos, de modo a poder seleccionar os elementos. A ideia fundamental deste algoritmo é, sempre que seja possível, poder seleccionar ou descartar elementos sem que para isso seja calculada a distância entre  $q$  e o elemento. Esta estratégia é usada na selecção/ descarte de conjuntos de elementos pertencentes a um dado agrupamento e em cada elemento dentro do agrupamento.

O algoritmo de pesquisa por alcance percorre a lista de agrupamentos e em cada iteração é calculada a distância  $d(q, c)$  do elemento de pesquisa  $q$  ao centro  $c$  do agrupamento  $(c, r, I)$  que estamos a analisar. Com base nesta distância, no raio de pesquisa  $r_q$  e no raio do agrupamento  $r$  são consideradas 6 situações (apresentadas nas figuras 4.13 e 4.14).

As primeiras 3 situações estão representadas na figura 4.13 e em todas verificamos que o centro faz parte do raio de pesquisa, ou seja,  $d(q, c) \leq r_q$ . Logo,  $c$  é automaticamente adicionado à lista de resultados da pesquisa.

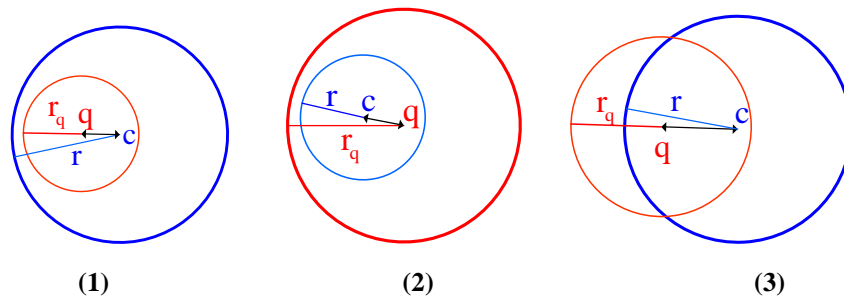


Figura 4.13 – Pesquisa no agrupamento abrange o centro deste

A primeira situação é verificada quando o espaço de pesquisa está completamente contido dentro agrupamento  $d(q, c) + r_q \leq r$  e contém o centro  $c$ . Neste caso, é necessário efectuar a pesquisa por alcance dentro do agrupamento, no subconjunto de elementos  $I$ . Como o espaço de pesquisa está completamente contido dentro do agrupamento não existe a necessidade de pesquisar em mais agrupamentos e a pesquisa termina.

Em relação à segunda situação, verificamos que o agrupamento está completamente contido no espaço de pesquisa, ou seja,  $d(q, c) + r \leq r_q$ . Desta forma, todos os elementos do agrupamento são adicionados à lista de resultados sem que seja necessário calcular a

medida de distância entre o elemento  $q$  e os elementos do agrupamento. De seguida é necessário chamar recursivamente o algoritmo para a restante lista de agrupamentos na medida em que o espaço de pesquisa pode intersectar mais agrupamentos.

Na terceira situação o espaço de pesquisa intersecta o agrupamento sem o conter completamente. Neste caso, é necessário efectuar a pesquisa dentro do agrupamento. Como o espaço de pesquisa pode intersectar mais agrupamentos é necessário chamar recursivamente o algoritmo para a restante lista de agrupamentos.

As últimas 3 situações estão representadas na figura 4.14 e em todas verificamos que o centro não faz parte do raio de pesquisa, ou seja,  $d(q,c) > r_q$ .

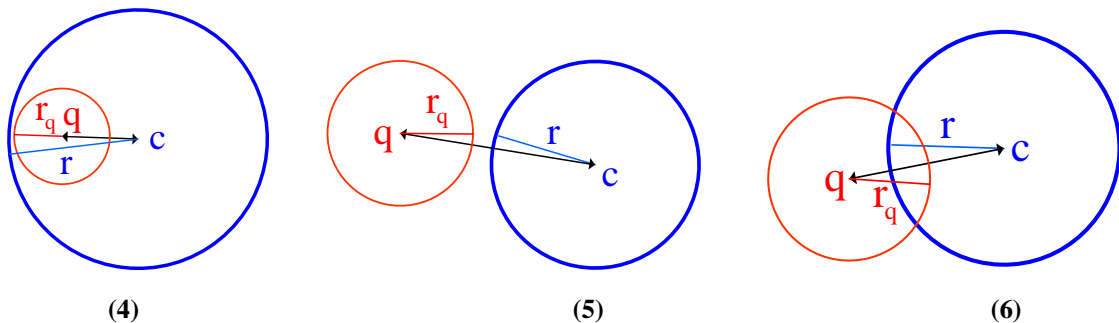


Figura 4.14 – Pesquisa no agrupamento não abrange o centro deste

Desta forma, a quarta situação é semelhante à primeira pois  $d(q,c) + r_q \leq r$ , no entanto,  $c$  não faz parte do conjunto de resultados. Neste caso, apenas é necessário realizar a pesquisa dentro do agrupamento e a pesquisa termina.

Em relação à quinta situação verificamos que o espaço de pesquisa não intersecta o agrupamento, ou seja,  $d(q,c) > r_q + r$ , logo nenhum elemento do agrupamento pertence ao conjunto de resultados. Neste caso, verificamos que todos os elementos do grupo são automaticamente descartados sem que seja calculada a distância entre os elementos e o elemento  $q$ . De seguida, é necessário continuar a pesquisa que é realizada através da chamada recursivamente do algoritmo para a restante lista de agrupamentos.

A última situação é semelhante à terceira com a diferença que  $c$  não faz parte da lista de resultados, ou seja, é necessário pesquisar dentro do agrupamento e continuar a pesquisar nos restantes agrupamentos da lista.

Dentro dum agrupamento  $(c_1, r, I_1)$ , a pesquisa por alcance trata, sempre que possível, de descartar ou seleccionar elementos sem ter que calcular a distância entre

esse elemento e o elemento  $q$ . Para tal são calculados os limites mínimos associados a cada centro pesquisado. Estes limites mínimos permitem descartar ou seleccionar elementos sem computar a distância em relação ao centro.

Por exemplo, na figura 4.15, considerando três elementos,  $c$  o centro do agrupamento,  $q$  o elemento de pesquisa e  $x$  o elemento a considerar se  $d(q,c) - r_q > d(x,c)$  então sabemos que  $d(x,q) > r_q$ , logo o elemento  $x$  pode ser descartado sem calcular nenhuma distância já que  $d(q,c)$  já foi calculado.

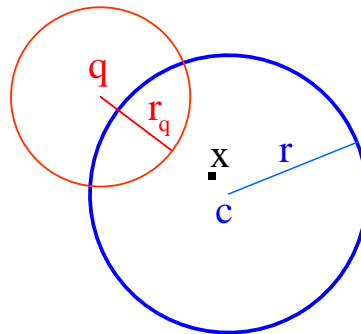


Figura 4.15 – Pesquisa dentro do agrupamento. O elemento  $x$  é descartado sem calcular  $d(q,x)$

Por outro lado, na figura 4.16 verificamos que se  $d(x,c) \leq -(d(q,c) - r_q)$  o elemento  $x$  é seleccionado sem ter que calcular nenhuma distância adicional.

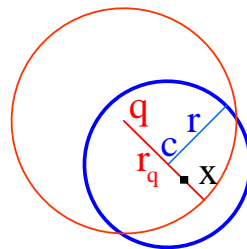


Figura 4.16 – Pesquisa dentro do agrupamento. O elemento  $x$  é seleccionado sem calcular  $d(q,x)$

Tendo como limite mínimo o valor  $d(q,c) - r_q$ , no algoritmo de pesquisa por alcance num dado agrupamento é criada uma sequência de limites mínimos com base nos centros pesquisados (hierarquia da estrutura) ou dos elementos no vector que serão usados para a pesquisa.

Como os elementos no vector estão ordenados por ordem decrescente em relação ao centro do agrupamento, sempre que, numa pesquisa por alcance dentro do vector, se encontra um elemento que pode ser descartado com base no primeiro limite mínimo

então todos os restantes elementos do vector podem ser ignorados sem cálculos adicionais da distância. O mesmo ocorre quando um elemento é seleccionado, ou seja, todos os outros são adicionados à lista da resposta sem cálculos adicionais.

### 4.2.3. *VPTree* – *Vantage Point Tree*

A estrutura de dados métrica designada de *VPTree* – *Vantage Point Tree* é apresentada em [24] e corresponde a uma estrutura estática, cuja partição do espaço é baseada em *pivots*. Neste caso, os vários objectos da base de dados encontram-se organizados numa árvore binária, com base na distância a um conjunto de elementos denominados *pivots*.

A raiz da árvore é um nó que contém um objecto da base de dados designado de *vantage point*, um raio, e as sub-árvores esquerda e direita. Os restantes objectos da base de dados pertencem às sub-árvores do nó de acordo com a sua distância ao objecto do nó. Neste caso, o raio é usado como critério para determinar em que sub-árvore o objecto da base de dados deve ser integrado. Desta forma, os objectos da base de dados próximos ao objecto guardado no nó, que não ultrapassem o valor do raio, encontram-se na sub-árvore esquerda enquanto que os restantes encontram-se na sub-árvore direita.

Cada sub-árvore do nó é também organizada numa árvore binária, ou seja, outra vez é criado um nó, seleccionando outro *vantage point*.

A figura 4.17 (b) retirada de [25] mostra um exemplo de uma *VPTree* associada ao espaço apresentado na figura 4.17(a), tendo como raiz o *pivot*  $u_{11}$ .

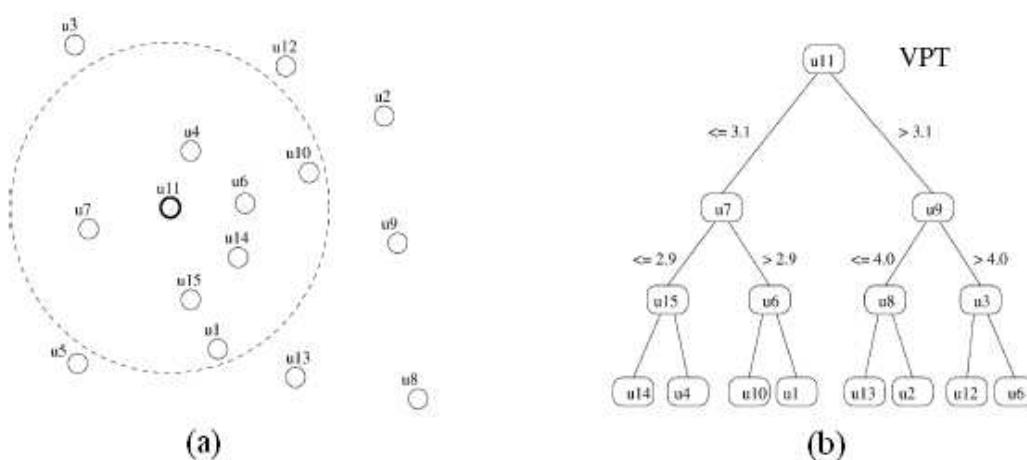


Figura 4.17 – Exemplo de uma *VPTree*

### 4.2.3.1. Algoritmo de construção

O algoritmo de construção da *VPTree* está representado na figura 4.18 e descreve a forma de organização dos vários objectos da base de dados  $B$  pela árvore.

```
Construção-VPT( $B$ )
  Se  $B = \emptyset$ 
    Retornar  $\emptyset$ ;
   $nó \leftarrow$  Criar Nó;
   $nó.vp \leftarrow$  Seleccionar Vantage Point( $B$ );
   $nó.raio \leftarrow$  Calcular Mediana( $nó.vp, B$ );
   $E \leftarrow \{b \in B - \{nó.vp\} \mid d(nó.vp, b) < nó.raio\}$ ;
   $D \leftarrow \{b \in B - \{nó.vp\} \mid d(nó.vp, b) \geq nó.raio\}$ ;
   $nó.arvE \leftarrow$  Construção-VPT( $E$ );
   $nó.arvD \leftarrow$  Construção-VPT( $D$ );
  Retornar nó;
```

Figura 4.18 – Construção de uma *VPTree*

Existem dois aspectos importantes neste algoritmo que dizem respeito ao valor do raio e à selecção do *vantage point*. O raio é seleccionado com base na mediana dos valores das distâncias do *vantage point* escolhido aos vários objectos da base de dados.

Uma aproximação simples para a escolha do *vantage point* consiste em escolher um objecto aleatório da base de dados.

Em [24], a selecção do *vantage point* baseia-se em seleccionar um conjunto de elementos candidatos. Estes são escolhidos aleatoriamente da base de dados que pretendemos dividir. De seguida, os vários elementos deste conjunto são percorridos para serem avaliados. O objectivo consiste em escolher o melhor candidato para ser o *vantage point*. Para realizar a avaliação de um candidato obtém-se outro conjunto com elementos aleatórios da base de dados. Para cada candidato é realizado o cálculo da mediana da distância entre o candidato e os vários elementos do conjunto e da variância.

Com base nesta avaliação aos candidatos é escolhido o melhor para *vantage point*, o qual corresponde ao que obteve o maior valor da variância associada.

De modo a otimizar a pesquisa é descrito em [24] uma optimização da *VPTree* descrita anteriormente, denominada de *VP<sup>s</sup> Tree*. A principal diferença em relação à estrutura anterior é que cada objecto mantém uma sequência de distâncias em relação aos *vantage points* de níveis superiores da árvore desde o nó onde se encontra o objecto até à raiz.

Esta informação pode ser usada aquando de uma pesquisa para descartar um elemento  $x$  sem que seja necessário realizar a computação da medida de distância entre

o elemento de pesquisa e  $x$ . Por exemplo, seja  $x$  o candidato a avaliar para a pesquisa  $(q, r_q)_d$ , o qual se encontra como elemento no nó de uma dada sub-árvore associada ao *pivot*  $p$ , se  $|d(q, p) - d(p, x)| > r_q$  então  $x$  é descartado já que  $d(q, x) > r_q$ . Por outro lado, se  $d(q, p) + d(p, x) \leq r_q$  então  $d(q, x) \leq r_q$  logo  $x$  é seleccionado.

#### 4.2.3.2. Algoritmo de pesquisa por alcance

Um algoritmo simples de pesquisa por alcance  $(q, r_q)_d$  referido em [25] começa por calcular a distância  $d(q, p)$  entre o elemento de pesquisa  $q$  e o *vantage point*  $p$  do nó correspondente à raiz da árvore. Se  $d(q, p) \leq r_q$  então o elemento correspondente ao *vantage point* faz parte do conjunto de resultados da pesquisa.

De seguida, é preciso verificar se é necessário pesquisar nas sub-árvores. Assim, se  $d(q, p) - r_q \leq r_p$ , sendo  $r_p$  o raio associado ao nó, então o algoritmo é realizado recursivamente para a sub-árvore esquerda do nó. Se  $d(q, p) + r_q > r_p$  então a recursividade é realizada na sub-árvore direita do nó. É de notar que, as duas sub-árvores podem ser pesquisadas se ambas as condições forem verificadas.

A descrição dos vários passos do algoritmo de pesquisa está na figura 4.19. Este recebe como parâmetros o nó correspondente à raiz da árvore, o elemento de pesquisa  $q$  e o raio  $r$  de pesquisa.

```

Pesquisa-VPTree(nó, q, r)
  Se nó = ∅
    Retornar ;
  dq ← d(nó.vp, q);
  Se dq ≤ r
    Adicionar nó.vp ao conjunto de resultados
  Se dq - r ≤ nó.raio
    Pesquisa-VPT(nó.arvE, q, r);
  Se dq + r > nó.raio
    Pesquisa-VPT(nó.arvD, q, r);

```

Figura 4.19 – Pesquisa por alcance numa *VPTree*

#### 4.2.4. GNAT – Geometric Near-neighbor Access Tree

A GNAT – Geometric Near-neighbor Access Tree referida em [27][25] é uma estrutura de dados métrica estática baseada em agrupamentos. A organização da informação é feita de forma hierárquica com base no domínio de Dirichlet, procurando reflectir a estrutura geométrica fundamental do conjunto de dados. Dado um conjunto de pontos  $\{x_1, \dots, x_n\}$ , o domínio de Dirichlet de  $x_i$  é o conjunto de pontos do espaço que estão mais próximos de  $x_i$  do que  $x_j$ , para qualquer  $(j \neq i)$ .

Para o nó superior da estrutura de dados GNAT são escolhidos vários elementos da base de dados como centros, e os restantes elementos são distribuídos por diversas regiões baseadas nos domínios de Dirichlet associados aos centros escolhidos. Assim, cada elemento fica associado ao centro mais próximo. Cada região é organizada recursivamente da mesma forma. Neste caso, verificamos que é criada uma estrutura em forma de árvore onde cada nó contém  $m$  filhos (regiões) associados aos  $m$  centros.

A figura 4.20 retirada de [25] mostra um exemplo do primeiro nível de uma estrutura de dados GNAT utilizando 4 centros em cada nível.

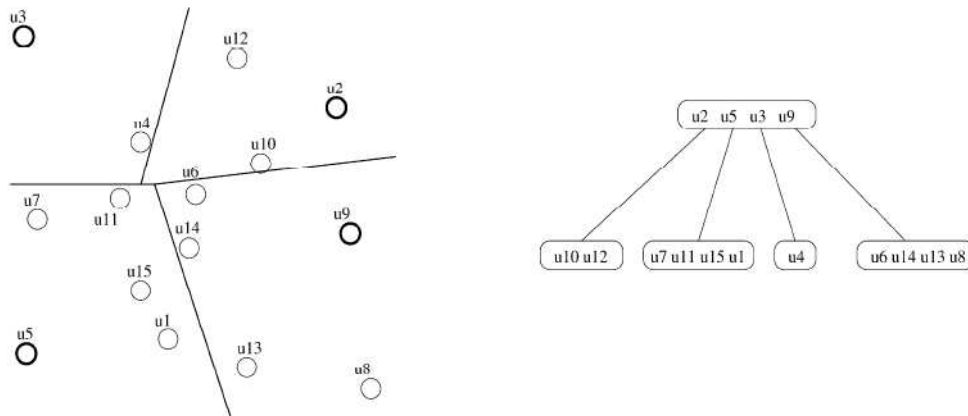


Figura 4.20 – Exemplo do primeiro nível de uma GNAT com 4 centros

Um aspecto importante está relacionado com a informação que é guardada em cada nó da árvore, a qual será utilizada para melhorar a eficiência da pesquisa por alcance. Cada nó tem associado uma tabela de dimensão  $m^2$  sendo  $m$  o número de centros de cada nó. Cada elemento da tabela  $t[i,j]$  com  $i,j \in [1..m]$  tem associado dois valores correspondentes às distâncias do centro  $i$  em relação ao ponto mais próximo e mais distante do outro centro  $j$ .

#### 4.2.4.1. Algoritmo de construção

A construção da estrutura de dados *GNAT*, para um conjunto de elementos  $B$ , é realizada nos seguintes passos:

1. São seleccionados  $m$  pontos centrais  $C = \{c_1, \dots, c_m\}$  para o novo nó, onde  $c_i \in B$ .
2. Os restantes pontos em  $B - C$  são associados aos pontos centrais do conjunto  $C$ . Cada elemento é associado ao centro mais próximos sendo criados  $m$  conjuntos. O conjunto de pontos associado ao centro  $c_i$  é designado de  $Dc_i$ .
3. Para cada par de pontos centrais  $(c_i, c_j)$  é criada uma entrada na tabela de intervalos onde é guardada para  $c_i$  os limites máximos e mínimos em relação a  $Dc_j$ . Esta entrada é designada de  $range(c_i, Dc_j) = [\min\_d(c_i, Dc_j), \max\_d(c_i, Dc_j)]$ .
4. É realizada a construção da árvore para cada conjunto  $Dc_i$  utilizando este mesmo algoritmo.

Um aspecto importante está relacionado com a escolha dos  $m$  centros em cada nó. A estratégia utilizada começa por escolher aleatoriamente  $3m$  elementos da base de dados que são os candidatos. De seguida, é escolhido um elemento aleatório deste conjunto. Depois, é escolhido do mesmo conjunto o ponto que se encontra mais longe do ponto escolhido anteriormente. De seguida, é escolhido o ponto mais longe dos dois anteriores e este critério de escolha é utilizado sucessivamente até serem encontrados os  $m$  pontos desejados. Neste caso, verificamos que é sempre seleccionado o ponto que está mais longe dos pontos escolhidos.

#### 4.2.4.2. Algoritmo de pesquisa por alcance

A pesquisa por alcance  $(q, r_q)_d$  percorre toda a árvore de modo a seleccionar os elementos que distam de  $q$  um valor igual ou inferior a  $r_q$ . Em cada nó é calculada a distância do elemento de consulta  $q$  com um dos centros do nó, seja  $c_i$ . Se  $c_i$  pertencer

ao raio da consulta então esse centro  $c_i$  é adicionado à lista de resultados. Com base na distância calculada  $d(q, c_i)$  e na tabela de intervalos associada a  $c_i$  podem ser descartados automaticamente alguns centros evitando a pesquisa nessas sub-árvores.

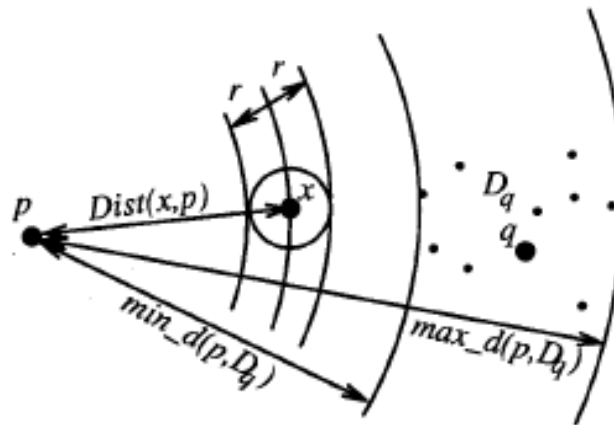


Figura 4.21 – Exemplo da utilização da tabela de intervalos

Por exemplo, tomando como base a figura 4.21 retirada de [27], os elementos  $p$  e  $q$  são dois centros,  $Dq$  corresponde ao conjunto de pontos do espaço que está mais próximo de  $q$ . No momento da pesquisa  $(x, r)_d$  no nó com centro em  $p$  existe uma entrada na tabela que relaciona  $p$  com o conjunto de pontos  $Dq$  e contém os limites  $min\_d(p, Dq)$  e  $max\_d(p, Dq)$ . Com base nesta informação verificamos que os elementos em  $Dq$  não fazem parte do conjunto de resultados da pesquisa se  $d(x, p) + r < min\_d(p, Dq)$ , logo foi possível eliminar um conjunto de elementos pertencente a  $Dq$  sem cálculos de distâncias adicionais.

Seja  $C$  o conjunto de elementos candidatos à pesquisa num dado nível, a pesquisa por alcance  $(q, r)_d$  numa estrutura GNAT é composta pelas seguintes fases:

1.  $C$  é inicializado com os centros do nó da raiz.
2. Obtém-se um ponto  $c \in C$  e se  $d(q, c) \leq r$  então  $c$  é adicionado à lista de resultados. O mesmo ponto central nunca é escolhido mais do que 1 vez.
3. Para o conjunto de centros  $c_j \in C$ , se  $[d(q, c) - r, d(q, c) + r] \cap range(c, Dc_j) = \emptyset$  então  $c_j$  é removido do conjunto  $C$ .
4. São repetidas as fases 2 e 3 até todos os pontos centrais do nó terem sido investigados.
5. Para cada ponto central  $c \in C$ , é realizada a mesma sequência de passos, partindo

da raiz da árvore associada a cada  $c$ .

#### **4.2.5. LAESA - Linear Approximating Eliminating Search Algorithm**

A LAESA – *Linear Approximating Eliminating Search Algorithm* referida em [26][25] é uma estrutura de dados métrica estática baseada em *pivots*, que usa uma matriz de distâncias pré-calculadas para poder tornar a pesquisa mais eficiente.

A ideia fundamental desta estrutura de dados consiste em usar  $k$  elementos da base de dados como *pivots*, onde cada um tem associado um vector de  $n$  distâncias calculadas, sendo  $n$  o número de elementos da base de dados. Ou seja, cada elemento  $x$  no vector  $d$  associado a um *pivot*  $p$ , denotado por  $d_p[x]$ , guarda a distância entre  $p$  e  $x$ ,  $d(p,x)$ .

##### **4.2.5.1. Algoritmo de construção**

No algoritmo de construção da estrutura de dados LAESA é realizada a selecção de  $k$  elementos da base de dados como *pivots* e é calculada a distância de cada *pivot* a todos os elementos da base de dados. Estas distâncias são guardadas para serem posteriormente usadas nas pesquisas.

Em relação à escolha dos *pivots* verificamos que o primeiro *pivot* é escolhido de forma aleatória. Sempre que um *pivot* é seleccionado é calculada a distância em relação a todos os elementos da base de dados. Para cada elemento da base de dados existe uma variável designada de acumulador que é incrementada com a distância do elemento ao último *pivot* escolhido. A selecção dos *pivots* seguintes é realizada com base nos acumuladores associados a cada elemento. Neste caso, verificamos que o *pivot* que é seleccionado corresponde ao elemento com o maior valor do acumulador, ou seja, o elemento que se encontra à maior distância dos *pivots* seleccionados anteriormente.

##### **4.2.5.2. Algoritmo de pesquisa por alcance**

A ideia fundamental do algoritmo de pesquisa por alcance  $(q, r_q)_d$  referido em [26]

reside em utilizar as distâncias já calculadas entre os *pivots* e um conjunto de elementos  $X$ , de modo a poder utilizar estas para calcular um limite inferior para a distância entre  $q$  e os elementos que ainda não foram descartados da pesquisa.

Como cada *pivot* mantém a distância em relação a todos os elementos da base de dados, notamos que ao relacionar o elemento de pesquisa  $q$  com um *pivot*  $p$  se um elemento da base de dados  $x \in X$  verificar a condição  $|d(q,p) - d(p,x)| > r_q$  então o elemento garantidamente pode ser descartado pois não faz parte da pesquisa (como podemos verificar na figura 4.2(c), com o *pivot*  $p_2$  . É necessário ter em conta que  $d(p,x)$  foi calculado na fase de inicialização da estrutura. Neste caso, verificamos que vários elementos podem ser descartados usando apenas o cálculo da medida de distância entre os *pivots* e o elemento  $q$ . Assim, no algoritmo de pesquisa por alcance os vários elementos da base de dados são percorridos e os que não puderem ser descartados considerando os vários *pivots* e a distância destes em relação ao elemento de pesquisa  $q$  são comparados directamente com  $q$  de modo a averiguar se fazem parte do conjunto de resultados.

#### **4.2.6. DSAT – Dynamic Spatial Approximation Tree**

A *DSAT – Dynamic Spatial Approximation Tree* referida em [28] é uma estrutura de dados métrica dinâmica e é baseada em agrupamentos.

Nesta estrutura os vários elementos são organizados numa árvore onde cada nó mantém um conjunto de elementos vizinhos, uma estampilha temporal associada ao momento de inserção do nó e um raio de cobertura referente à distância do nó ao elemento mais distante presente no conjunto de vizinhos. O conjunto de elementos vizinhos corresponde aos filhos do nó, pelo que é necessário estipular a aridade máxima da árvore, a qual corresponde à capacidade máxima de vizinhos num nó. Cada vizinho está representado por um nó e é construído da mesma forma.

A figura 4.22 adaptada de [25] mostra um exemplo de uma *DSAT* com aridade máxima igual a 3 e raiz no nó *u11*.

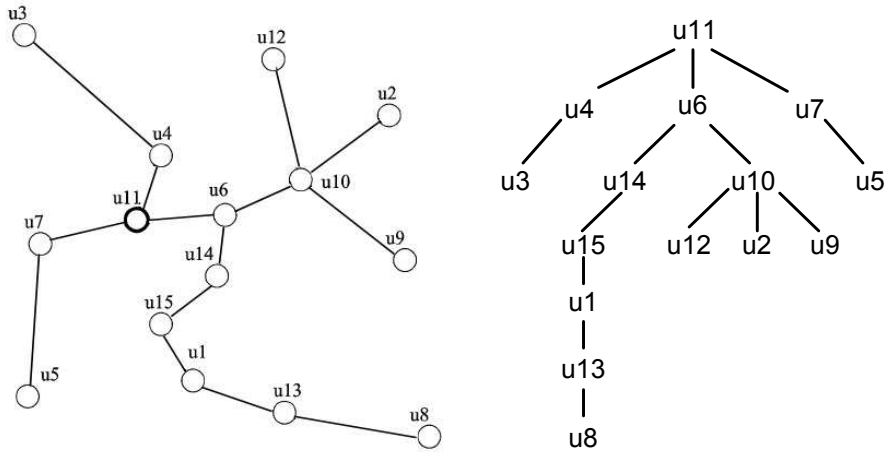


Figura 4.22 – Exemplo de uma *DSAT* no espaço e da sua representação hierárquica

#### 4.2.6.1. Algoritmo de construção

Em relação à construção verificamos que é realizada por sucessivas operações de inserção. A *DSAT* pode ser construída a partir de um único nó, sem elementos e com um raio de cobertura igual a 0. A estrutura da árvore depende da ordem de inserção.

#### 4.2.6.2. Algoritmo de inserção

A operação de inserção de um novo elemento começa pela raiz da árvore. O elemento é adicionado ao nó como uma folha se a distância em relação este for mais próxima do que em relação a qualquer filho e se o número de filhos não tiver excedido a capacidade do nó. A figura 4.23, adaptada de [25], descreve esta situação de inserção de um novo elemento na *DSAT* da figura 4.22. Neste caso, verificamos a vermelho o caminho percorrido desde a raiz *u11* até ao nó *u4* onde foi inserido. O elemento *el* foi inserido no nó *u4* porque verificamos que é o nó mais próximo, ou seja, a distância de *el* em relação ao filho *u3* é maior do que em relação ao nó e porque *u4* não excedeu a capacidade do nó.

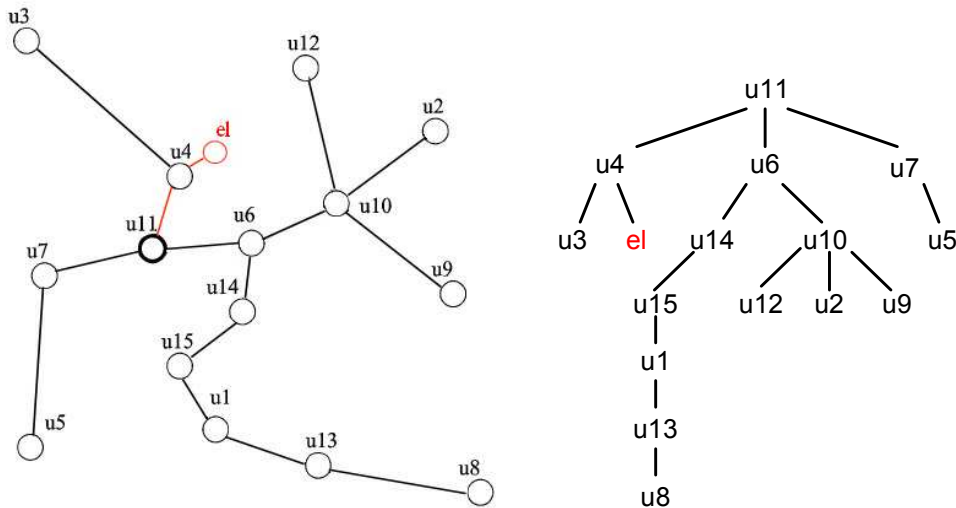


Figura 4.23 – Exemplo de uma inserção numa *DSAT*

No caso em que o nó excede a capacidade, é escolhido o filho mais próximo ao elemento a inserir e o algoritmo é realizado de forma recursiva para a sub-árvore associada ao filho seleccionado. A figura 4.24 adaptada de [25] ilustra um caso destes na inserção de um novo elemento *el* na árvore *DSAT* apresentada na figura 4.22. Apesar de *el* estar mais perto de *u10*, notamos que foi inserido em *u2* porque *u10* tem 3 filhos que corresponde à capacidade máxima de vizinhos num nó. Desta forma, foi escolhido o vizinho mais próximo de *el*, que corresponde ao nó *u2*.

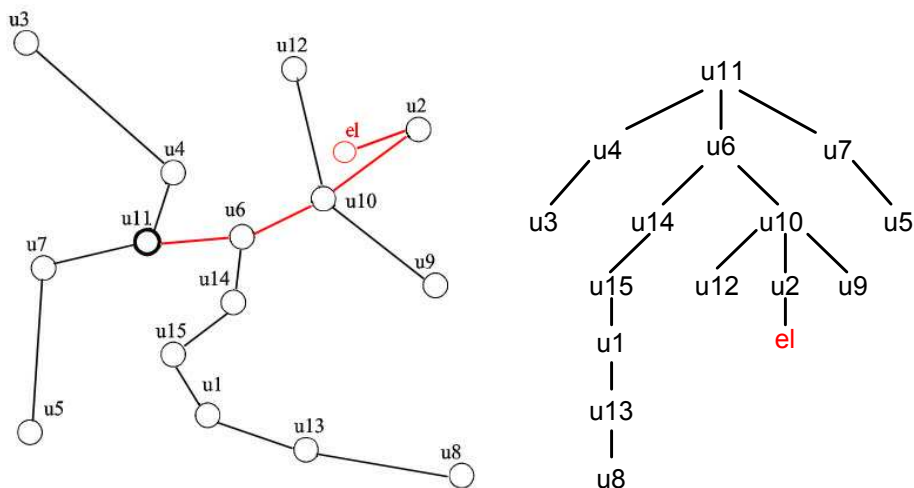


Figura 4.24 – Exemplo de inserção na *DSAT* num nó que excedeu a capacidade

É de notar que apesar do elemento *el* estar mais próximo de *u10* que de qualquer um dos seus vizinhos, *el* passa a ser vizinho de um vizinho de *u10*.

Depois de encontrado o nó para o elemento, este é inserido no final da lista de vizinhos pertencentes ao nó e associamos a estampilha temporal ao elemento. A estampilha temporal corrente é incrementada. Como os vários elementos são inseridos sempre no final da lista verificamos que os elementos estão ordenados de forma crescente pela estampilha temporal. Por exemplo, na figura 4.24 podemos notar que no caso dos vizinhos  $u_{12}$ ,  $u_2$ , e  $u_9$ , garantidamente o elemento  $u_{12}$  foi inserido primeiro que  $u_2$  e este primeiro que  $u_9$ .

#### **4.2.6.3. Algoritmo de remoção**

O primeiro passo do algoritmo de remoção da *DSAT* consiste em encontrar o elemento a remover na árvore. Uma solução elegante para este problema consiste em realizar uma pesquisa por alcance de raio 0, tendo como elemento de consulta o elemento a remover. Desta forma, serão pesquisados todos os sítios possíveis da árvore onde o elemento a remover poderá estar localizado. Se o elemento a remover for uma folha, então este é removido sem nenhum custo. Caso contrário é necessário reinserir os vizinhos do nó que foi removido. Em [38] são apresentadas 4 alternativas para tratar este caso. Na primeira e segunda alternativa a sub-árvore que contém o nó a remover é separada da árvore principal e é reinserida de novo (toda ou por partes) a partir da raiz da árvore. Na terceira opção, a sub-árvore que foi afectada é reconstruída como se o elemento que foi removido nunca tivesse sido inserido na árvore. Esta opção (ao contrário das duas primeiras) garante a qualidade da árvore depois de sucessivas remoções[38]. A última alternativa consiste em marcar o elemento como removido. Desta forma, de modo a evitar a degradação das pesquisas é necessário reconstruir a árvore periodicamente.

Em relação à operação de remoção, é necessário ter em conta que a maioria dos nós da árvore são folhas, especialmente quando a aridade da árvore é alta. Desta forma, existe uma motivação para o uso de aridades maiores quando a operação de remoção é considerada.

#### 4.2.6.4. Algoritmo de pesquisa por alcance

Em [28] é apresentado o algoritmo de pesquisa por alcance na estrutura *DSAT*. Sendo  $q$  o elemento de pesquisa e  $r$  o raio desta verificamos que o algoritmo, tendo como alvo a pesquisa, procede como se fosse inserir o elemento  $q$  utilizando uma tolerância de  $\pm r$ .

O algoritmo de pesquisa por alcance percorre os nós da árvore começando pela raiz. Para cada nó calcula a distância do elemento  $p$ , que representa o nó, ao elemento de pesquisa  $q$ . Se  $d(p, q) > r_p + r$ , sendo  $r_p$  o raio associado ao nó, então o algoritmo termina para essa árvore, pois neste caso sabemos que não existe nenhum elemento na árvore que intersecte a consulta já que  $r_p$  é a distância do centro  $p$  ao seu vizinho  $x$  mais distante, logo temos  $d(p, q) > d(p, x) + r$ . Desta forma,  $\forall y$  tal que  $y$  é vizinho de  $p$  e  $y \neq x$  sabemos que  $d(p, y) < d(p, x)$  logo a  $d(p, q) > d(p, y) + r$  e  $d(p, q) - d(p, y) > r$  e por conseguinte a  $d(q, y) > r, \forall y$ .

Caso contrário, se o elemento que representa o nó faz parte da pesquisa então é adicionado ao conjunto de resultados. De seguida é necessário pesquisar nos vizinhos relevantes do nó. Desta forma, seja  $\{b_1, \dots, b_k\}$  o conjunto de vizinhos de um nó ordenado de forma crescente pela estampilha temporal, verificamos que a pesquisa de elementos relevantes na sub-árvore representada por  $b_i$  é realizada recursivamente nesta se  $d(q, b_i) \leq \min(d(q, b_1), \dots, d(q, b_{i-1})) + 2r$ . Desta forma, verificamos que o algoritmo de pesquisa tem em conta que entre a inserção de  $b_i$  e  $b_{i+j}$  podem ter existido elementos que foram inseridos na sub-árvore representada por  $b_i$  pelo facto de  $b_{i+j}$  não existir e que  $d(o, b_{i+j}) < d(o, b_i) \forall o$  pertencente à sub-árvore de  $b_{i+j}$ .

Outro aspecto que é considerado na pesquisa está relacionado com as estampilhas temporais. Esta informação é utilizada com o objectivo de reduzir o trabalho de pesquisa dentro de vizinhos mais antigos. Supondo que no momento da pesquisa se verifica que  $d(q, b_i) > d(q, b_{i+j}) + 2r$  então é necessário pesquisar na sub-árvore  $b_i$  por ser mais antiga e porque elementos inseridos antes de  $b_{i+j}$  (próximos deste) podem ter sido inseridos em  $b_i$  pelo facto de  $b_{i+j}$  não existir na árvore. Neste caso, na pesquisa em  $b_i$  um nó que tenha uma estampilha temporal maior que a de  $b_{i+j}$  pode ser descartado.

A figura 4.25 descreve o algoritmo de pesquisa por alcance da *DSAT*. Este recebe como parâmetros a raiz da árvore, o elemento de pesquisa  $q$ , um raio de pesquisa  $r_q$  e uma estampilha temporal  $t$ . O algoritmo de pesquisa é invocado com a estampilha temporal  $t = \infty$ . Neste algoritmo  $nó.t$  representa a estampilha temporal do elemento que representa o nó,  $nó.r$  corresponde ao raio de cobertura,  $nó.N$  representa o conjunto de vizinhos do nó ordenados pela estampilha temporal.

```

Pesquisa-DSAT( $nó, q, r_q, t$ )
  Se  $nó.t < t \wedge d(nó, q) \leq nó.r + r_q$ 
    Se  $d(nó, q) \leq r_q$ 
      Adicionar nó à lista de resultados;
     $d_{\min} \leftarrow \infty$ ;
    for  $b_i \in nó.N$ 
      if  $d(b_i, q) \leq d_{\min} + 2r_q$ 
         $k \leftarrow \min\{ j > i, d(b_j, q) > d(b_i, q) + 2r_q \}$ 
        Pesquisa-DSAT( $b_i, q, r_q, b_k.t$ );
       $d_{\min} \leftarrow \min\{d_{\min}, d(b_i, q)\}$ 

```

Figura 4.25 – Operação de pesquisa na *DSAT*

#### 4.2.7. *HDSAT – Hybrid Dynamic Spatial Approximation Tree*

A *HDSAT – Hybrid Dynamic Spatial Approximation Tree* referida em [28] é uma estrutura de dados métrica dinâmica que deriva da *DSAT*. Esta estrutura permite melhorar os tempos de pesquisa da *DSAT*, usando para isso técnicas de *pivot*.

Existem duas versões designadas de *HDSAT1* e *HDSAT2*. A diferença entre as duas encontra-se na escolha dos *pivots*. Estes são escolhidos durante a inserção de um novo elemento na estrutura. Desta forma, verificamos que cada nó fica associado com um conjunto de *pivots*.

Nas pesquisas verificamos que a informação dos *pivots* de cada nó é usada para restringir o espaço da pesquisa. Neste caso, quando o algoritmo de pesquisa entra num nó, o elemento de pesquisa é comparado com os *pivots* do nó e com base nesta informação podem ser descartados filhos pertencentes ao nó sem que seja necessário calcular a função de distância entre o elemento de consulta e esses filhos.

### 4.2.7.1. HDSAT1 – Hybrid Dynamic Spatial Approximation Tree 1

Em relação à *HDSAT1* verificamos que são considerados os vários antecessores de cada nó como os seus *pivots*. Os antecessores correspondem aos vários nós da árvore que foram percorridos aquando da inserção de um novo elemento. Neste caso verificamos que o algoritmo de inserção é semelhante ao descrito na *DSAT*. A única diferença está relacionada com os antecessores.

Para inserir o elemento  $x$  é necessário percorrer a árvore até conseguir a posição para o novo elemento. Sempre que entramos num nó representado por  $p$ , é criado o par  $(p, d(p, x))$  e é adicionado a uma lista que será guardada quando o elemento finalmente é inserido num dado nó.

A figura 4.26 adaptada de [25] demonstra a lista de *pivots* com as respectivas distâncias para o nó  $u2$  considerando  $u11$  como raiz.

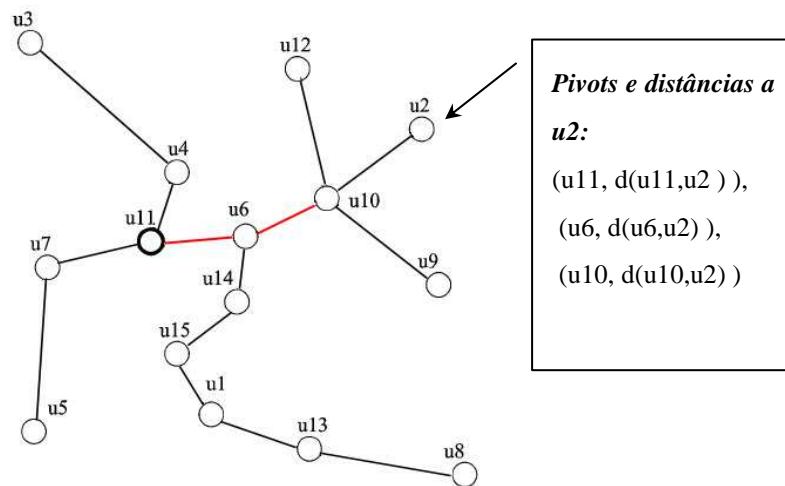


Figura 4.26 – Conjunto de *pivots* e respectivas distâncias do nó  $u2$  de uma *HDSAT1*

Em relação à pesquisa por alcance verificamos que é semelhante à realizada na *DSAT* no entanto é considerada a informação dos *pivots*. Esta informação é utilizada para decidir se é necessário realizar a pesquisa nos elementos vizinhos de cada nó.

Desta forma, quando o algoritmo de pesquisa por alcance  $(q, r_q)_d$  entra num determinado nó  $p_k$  é necessário saber se é necessário fazer a pesquisa em cada vizinho  $x$  de  $p_k$ .

Neste momento da pesquisa, já temos calculado todas as distâncias  $d(q, p_i)$ , onde

$p_i$  são os antecessores de  $x$ . Para além disso, no nó  $x$  temos  $\{(p_1, d(x, p_1)), \dots, (p_k, d(x, p_k))\}$ , sendo  $p_i$  os antepassados de  $x$ . Logo, é possível calcular  $\max_{p_i} |d(x, p_i) - d(q, p_i)|$  sem cálculos adicionais. A este valor denotamos por  $\mathcal{D}(x, q)$ . Com base neste valor, é possível saber quais são os vizinhos  $x$  que devemos pesquisar, já que são aqueles cujo  $\mathcal{D}(x, q) \leq x.r + r_q$ , onde  $x.r$  corresponde ao raio de cobertura do nó  $x$ . O valor  $\mathcal{D}(x, q)$  é um limite inferior para  $d(x, q)$ . Logo, se  $\mathcal{D}(x, q) > x.r + r_q$ , pode-se descartar  $x$  da pesquisa e evitar a computação de  $d(x, q)$ , já que se  $\mathcal{D}(x, q) > x.r + r_q$  então  $d(x, q) > x.r + r_q$ . Assim, ao contrário do algoritmo de pesquisa na *DSAT* podem ser evitadas algumas computações da medida de distância entre o elemento de pesquisa e os vários elementos vizinhos do nó que estamos a analisar.

Em [39] é descrito o algoritmo de remoção da *HDSAT1* que corresponde a uma adaptação da terceira alternativa do algoritmo de remoção da *DSAT* (ver secção 4.2.6.3). O algoritmo de remoção da *HDSAT1* necessita de ter em conta a existência dos *pivots* de cada nó, pois é necessário remover os *pivots* inválidos. Neste caso, a informação dos *pivots* pode ser aproveitada para reduzir o custo do algoritmo de remoção.

#### 4.2.7.2. *HDSAT2 – Hybrid Dynamic Spatial Approximation Tree 2*

Na estrutura *HDSAT2*, verificamos que em cada nó são considerados mais *pivots* que na *HDSAT1*. O algoritmo de inserção de um novo elemento  $x$  é semelhante ao da *HDSAT1* no que diz respeito aos antecessores. Ou seja, ao entrar num nó é criado o par  $(a, d(a, x))$  e este é adicionado no final da lista de *pivots* correspondentes a  $x$ . Para além de considerar o  $a$  como *pivot*, sempre que seja necessário aplicar o algoritmo recursivamente para uma sub-árvore representada por  $b$ , o qual é o vizinho de  $a$  mais próximo do elemento  $x$ , é comparada a estampilha temporal de  $b$  com a dos restantes vizinhos e se a estampilha temporal for menor que a do elemento  $b$  então esse elemento vizinho é adicionado como *pivot*.

Concluindo, seja  $A(x)$  o conjunto de nós antecessores, que foi percorrido aquando da inserção de um elemento  $x$ , o conjunto de *pivots* de  $x$  denotado por  $P(x)$  está definido da seguinte forma  $P(x) = A(x) \cup \{b: b \in p_i.N \wedge p_i, p_{i+1} \in A(x) \wedge p_{i+1} \in p_i.N \wedge b.t < p_{i+1}.t\}$ ,

onde  $p_i.N$  corresponde ao conjunto de nós vizinhos de  $p_i$  e  $b.t$  corresponde à estampilha temporal do nó  $b$ .

Em relação à pesquisa por alcance verificamos que o algoritmo apenas difere na forma como é calculado  $\mathcal{D}(b,q)$ .

O algoritmo de remoção da *HDSAT2* é semelhante ao da *HDSAT1*. No entanto, neste caso o custo do algoritmo de remoção pode ser ainda mais reduzido pois na *HDSAT2* são utilizados mais *pivots*.

### 4.3. Análise comparativa

Existem poucos trabalhos que comparam estas estruturas de dados em domínios de aplicação. No entanto, em [7] foi realizada uma avaliação de várias estruturas de dados métricas no domínio das linguagens (dicionários de Inglês, Francês, Alemão, Italiano, Português e Espanhol). As estruturas de dados métricas utilizadas em [7] foram a *GNAT*, *HDSAT1*, *LAESA*, *LC*, *VPtree* e *RLC* e em [31] foram a *RLC* e *VPtree*.

Em [7] a métrica utilizada para medir a semelhança entre palavras foi a *Levenshtein* e em [31] foi uma extensão da função de edição.

Nas experiências realizadas em [7] com o menor raio de pesquisa a *LAESA* foi a estrutura de dados que obteve o melhor desempenho em todos os dicionários, seguida da *RLC*. Com o segundo raio de pesquisa, a *LAESA* continuou a ser a estrutura mais eficiente em todos os dicionários exceptuando o dicionário de Português onde foi a *RLC* que obteve os melhores resultados. Com o maior raio de pesquisa, a performance da *LAESA* degradou-se bastante e verificamos que a *LC* e *RLC* foram as estruturas mais eficientes. Em relação às duas estruturas de dados métricas dinâmicas verificamos que a *RLC* foi, em todos os casos, mais eficiente que a *HDSAT1*.

Nas experiências realizadas em [31], a *VPtree* obteve o melhor desempenho com o menor raio de pesquisa, mas a *RLC* foi a melhor em todos os casos, com os outros raios de pesquisa.

## 4.4. Conclusão

Neste capítulo foram descritas várias estruturas de dados métricas, propostas com o objectivo de melhorar a eficiência das pesquisas por similaridade em espaços métricos.

O conjunto de estruturas de dados métricas que foi estudado é composto pela *LC*, *RLC*, *VPTree*, *GNAT*, *LAESA*, *DSAT*, *HDSAT1*, *HDSAT2*. Deste conjunto, existem algumas estruturas de dados métricas que são dinâmicas tais como a *RLC*, *DSAT* e derivadas desta última (*HDSAT1*, e *HDSAT2*).

Nesta tese vamos implementar e avaliar a aplicabilidade e eficiência das várias estruturas de dados métricas estudadas no domínio das imagens de rosto em diferentes bases de dados, com diferentes funções de semelhança.

É de salientar que para este tipo de aplicações (registos criminais, registos de passaportes, registos de cartas de condução, entre outras) as estruturas de dados métricas dinâmicas são as mais relevantes já que as bases de dados de imagens de rosto não são estáticas.

Não foram encontrados trabalhos na área de reconhecimento facial que tivessem usado estruturas de dados métricas para agilizar as pesquisas.



## **Parte II – Pacote de estruturas de dados métricas**



## Capítulo 5. Pacote de estruturas de dados métricas

Neste capítulo será descrito o pacote de estruturas de dados métricas que foi implementado nesta dissertação. Este pacote foi implementado em Java em paralelo com Francisco Costa que irá avaliar a aplicabilidade e eficiência das estruturas de dados métricas na pesquisa por alcance de músicas semelhantes.

As estruturas de dados métricas implementadas foram a *LAESA*, *GNAT*, *VPTree*, *DSAT*, *HDSAT1*, *HDSAT2*, *LC* e *RLC*.

Este capítulo está organizado em 4 secções. Na primeira é descrita a organização do pacote. Nas seguintes duas secções são descritas brevemente as interfaces e classes disponibilizadas no pacote. E por último é apresentada a conclusão.

### 5.1. Organização do pacote

A figura 5.1 descreve a organização das diferentes interfaces e classes que compõem o pacote de estruturas de dados métricas designado de “dataStructures”.

A interface `IMetricDataStructure<E>` representa uma estrutura de dados métrica de elementos do tipo genérico `E`. Esta interface contém todas as operações comuns nas diferentes estruturas de dados, tais como a pesquisa por alcance. A interface `IDynamicDataStructure<E>` corresponde a uma extensão da interface `IMetricDataStructure<E>` e representa uma estrutura de dados métrica dinâmica de elementos do tipo genérico `E`. Neste caso, foram adicionadas as operações de inserção e remoção de um elemento na estrutura. Desta forma, todas as estruturas de dados métricas dinâmicas tais como a *DSAT*, *HDSAT1*, *HDSAT2* e *RLC* implementam a interface `IDynamicDataStructure<E>`.

As estruturas de dados métricas baseiam-se no conceito de espaço métrico, logo é necessário definir a métrica que será usada para medir a semelhança entre os vários elementos do tipo `E`. Assim sendo, foi criada a classe abstracta

MetricDataStructure<E>, que implementa a interface IMetricDataStructure<E>, e cujo construtor recebe como argumento a métrica que será usada pela estrutura de dados. Esta métrica está representada na interface IMetricComparator<E>, a qual contém a operação que permite computar a distância entre dois elementos do mesmo tipo genérico E.

Todas as estruturas de dados métricas são classes que estendem a classe abstracta MetricDataStructure<E>

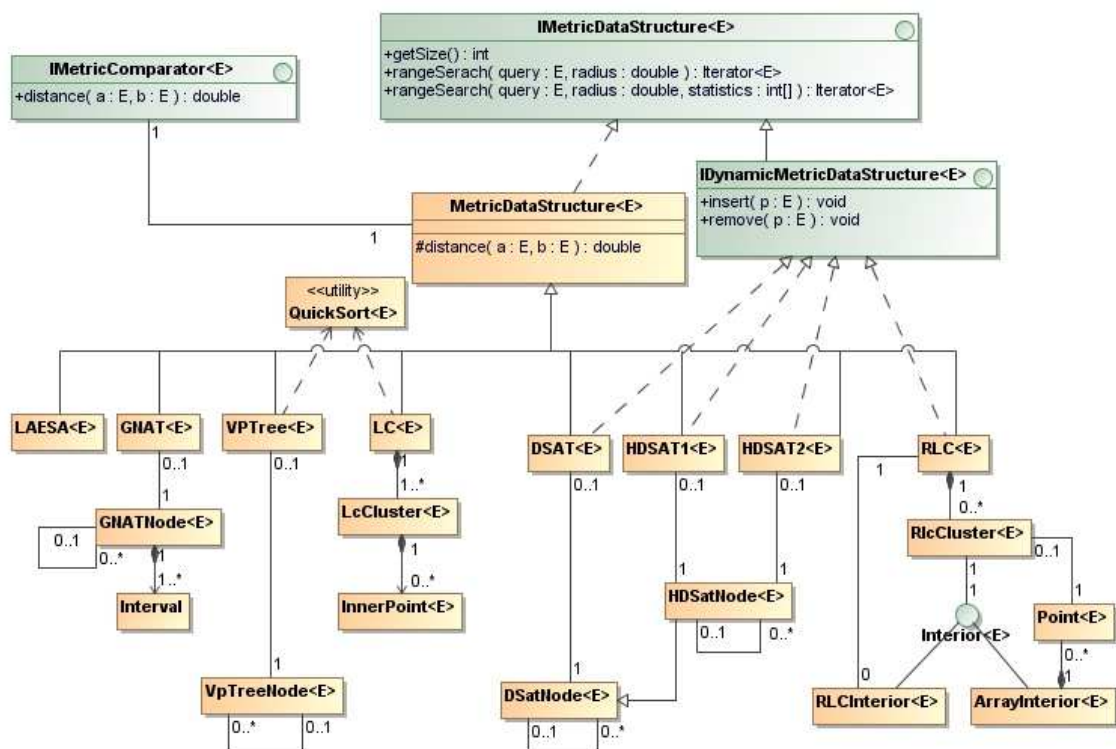


Figura 5.1 – Diagrama de classes do pacote “dataStructures”

## 5.2. Descrição das interfaces

Nesta secção serão descritos os métodos e os respectivos parâmetros das interfaces IMetricDataStructure<E>, IDynamicMetricDataStructure<E> e IMetricComparator<E>.

A figura 5.2 descreve a interface IMetricDataStructure<E> que é composta pelas operações comuns a todas as estruturas de dados métricas.

```

public interface IMetricDataStructure<E>
{
    int getSize();
    Iterator<E> rangeSearch(E query,double radius);
    Iterator<E> rangeSearch(E query,double radius,int[] statistics);
}

```

Figura 5.2 – Interface `IMetricDataStructure<E>`

O método `getSize()` devolve o número de elementos presentes na estrutura de dados métrica.

Para realizar uma pesquisa por alcance utilizamos o método `rangeSearch(E query,double radius)` que tem como parâmetros de entrada o elemento de consulta (*query*) e o raio de pesquisa (*radius*). Este método devolve um iterador de elementos do tipo `E`, os quais são todos os resultados encontrados que fazem parte da pesquisa efectuada.

O terceiro método desta interface é igual ao segundo mas tem mais um parâmetro, que é utilizado para fins estatísticos. Neste caso, na primeira posição do vector `statistics` é contabilizado o número de computações efectuadas na operação de pesquisa. É com base neste parâmetro que depois são feitas as comparações de eficiência das diferentes estruturas de dados métricas (ver secção 8.3).

A figura 5.3 descreve a interface `IDynamicMetricDataStructure<E>` que representa uma estrutura de dados métrica dinâmica.

```

public interface IDynamicMetricDataStructure<E> extends IMetricDataStructure<E>
{
    void insert(E p);
    void remove(E p);
}

```

Figura 5.3 – Interface `IDynamicMetricDataStructure<E>`

Como podemos verificar pela figura, esta interface estende a `IMetricDataStructure<E>` e contém mais dois métodos para a inserção e remoção de elementos do tipo `E`. Tanto o primeiro como o segundo método recebem como parâmetro o elemento que queremos inserir ou remover, respectivamente.

A figura 5.4 descreve a interface `IMetricComparator<E>` que representa uma

métrica. Todas as estruturas de dados necessitam de uma métrica que será usada para medir a similaridade entre os vários elementos que serão inseridos na estrutura. Como será visto na secção seguinte, esta métrica é passada no momento da criação da estrutura de dados métrica.

Como podemos verificar pela figura 5.4 a interface `IMetricComparator<E>` contém o método `distance(E a, E b)` que calcula a distância/semelhança entre dois elementos do mesmo tipo `E`, os quais são dados como parâmetros de entrada e devolve um valor real.

```
public interface IMetricComparator<E>
{
    double distance(E a, E b);
}
```

Figura 5.4 – Interface `IMetricComparator<E>`

### 5.3. Descrição das classes

Cada estrutura de dados foi implementada numa classe com base na descrição do capítulo 4. Para cada classe disponível no pacote serão descritos os construtores das classes e serão apresentadas algumas decisões de implementação tomadas. Também será apresentada a relação com as diversas interfaces e classes auxiliares que foram utilizadas na sua implementação.

#### 5.3.1. LAESA

A figura 5.5 descreve a implementação da estrutura de dados *LAESA*.

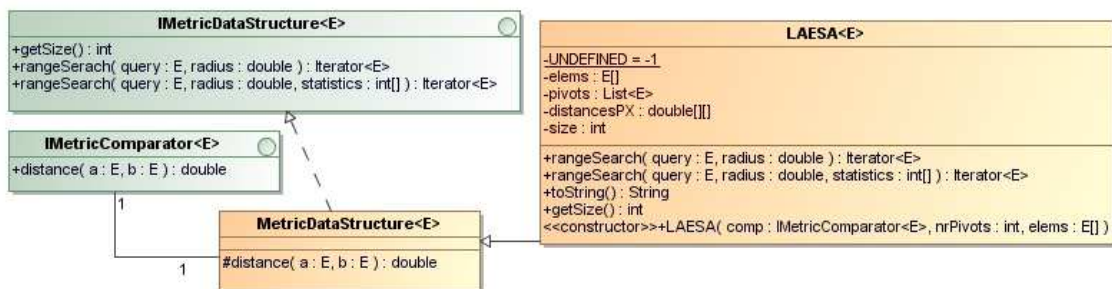


Figura 5.5 – Classe *LAESA*

A declaração do construtor da *LAESA* tem 3 parâmetros e é definida como:

- **public LAESA(IMetricComparator<E> comparator, int nrPivots, E[] elems)**

O primeiro parâmetro diz respeito à métrica que será usada na construção da estrutura de dados, o segundo corresponde ao nº de *pivots* que será usado, e o último é o conjunto dos elementos que serão inseridos na estrutura de dados.

### 5.3.2. *VPTree*

A figura 5.6 descreve a implementação da estrutura de dados *VPTree*.

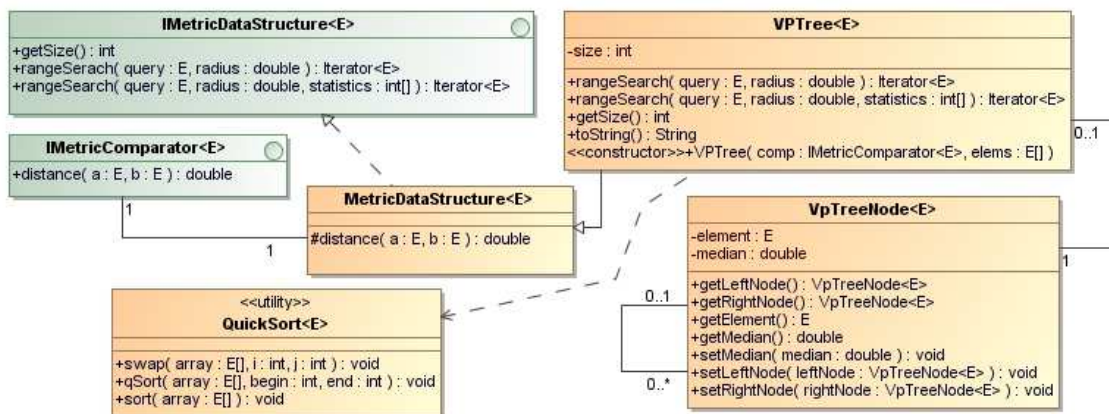


Figura 5.6 – Classe *VPTree*

Neste caso foi criado um construtor com 2 parâmetros de entrada que correspondem respectivamente à métrica e ao conjunto de elementos da base de dados. A declaração do construtor da *VPTree* é definido da seguinte forma:

- **public VPTree(IMetricComparator<E> comparator, E[] elems)**

Em relação ao algoritmo de construção da *VPTree* foi utilizada a aproximação da selecção aleatória do *vantage point*.

### 5.3.3. *LC*

A figura 5.7 descreve a implementação da estrutura de dados *LC*.

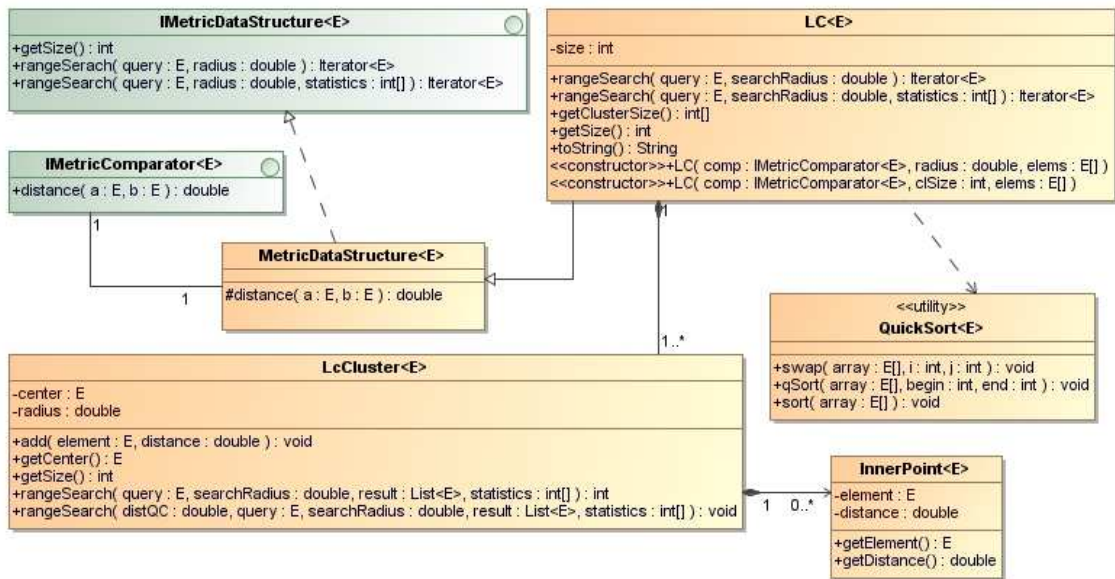


Figura 5.7 – Classe *LC*

Neste caso foram criados dois construtores definidos como:

- **public LC(IMetricComparator<E> comparator, double radius, E[] elems)**
- **public LC(IMetricComparator<E> comparator, int clusterInteriorSize, E[] elems)**

A primeira declaração é usada para criar uma *LC* com agrupamentos de raio fixo. Neste caso, o primeiro parâmetro diz respeito à métrica, o segundo define o raio de cada agrupamento e o terceiro corresponde ao conjunto de elementos.

Em relação à selecção do centro de cada agrupamento, neste construtor, é escolhido o elemento que minimiza as distâncias aos centros dos agrupamentos anteriores.

A segunda declaração constrói uma *LC* com agrupamentos com um número fixo de elementos. Desta forma, a diferença em relação à primeira declaração encontra-se no parâmetro 2 que, neste caso, corresponde ao número de elementos máximo permitido no interior de um agrupamento. A heurística utilizada para a escolha do centro é a que maximiza a soma das distâncias aos centros dos agrupamentos anteriores.

### 5.3.4. GNAT

A figura 5.8 descreve a implementação da estrutura de dados *GNAT*.

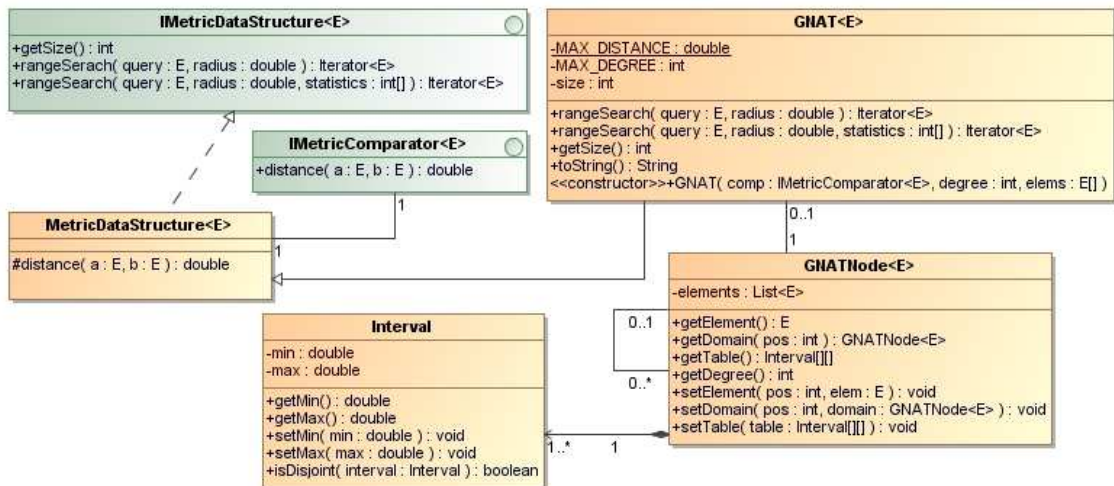


Figura 5.8 – Classe *GNAT*

A declaração do construtor da *GNAT* é definida da seguinte forma:

- **public GNAT(IMetricComparator<E> comparador, int degree, E[] elems)**

Como podemos verificar a *GNAT* tem um construtor com 3 parâmetros onde o primeiro corresponde à métrica, o segundo define o número máximo de centros de cada nó, e o terceiro corresponde ao conjunto de elementos da base de dados.

### 5.3.5. *DSAT* e variantes

A figura 5.9 descreve a implementação da estrutura de dados *DSAT* e das suas variantes.

Para a *DSAT* foram criados dois construtores, definidos como:

- **public DSAT(IMetricComparator<E> comparador, int maxArity, E[] elems)**
- **public DSAT(IMetricComparator<E> comparador, int maxArity)**

A primeira declaração tem 3 parâmetros que correspondem respectivamente à métrica, ao número máximo de elementos vizinhos por nó e ao conjunto de elementos. A segunda declaração não tem o parâmetro correspondente ao conjunto de elementos e é usado para criar uma *DSAT* vazia.

A *HDSAT1* e *HDSAT2* têm dois construtores com os mesmos parâmetros da classe *DSAT* e são definidos como:

- **public HDSAT1(IMetricComparator<E> comparador, int maxArity, E[] elems)**

- **public HDSAT1(IMetricComparator<E> comparator, int maxArity)**
- **public HDSAT2(IMetricComparator<E> comparator, int maxArity, E[] elems)**
- **public HDSAT2(IMetricComparator<E> comparator, int maxArity)**

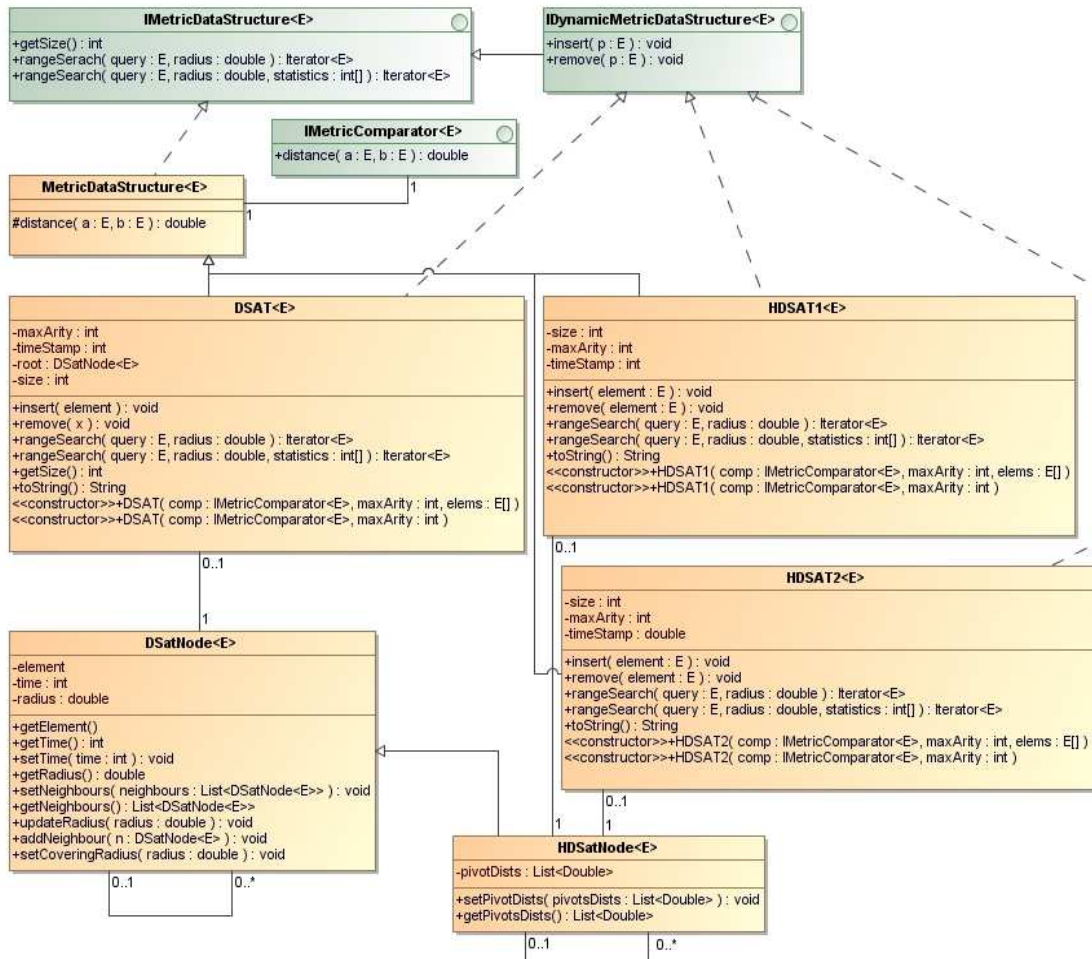


Figura 5.9 – Classes *DSAT*, *HDSAT1* e *HDSAT2*

### 5.3.6. RLC

A figura 5.10 descreve a implementação da estrutura de dados *RLC*.

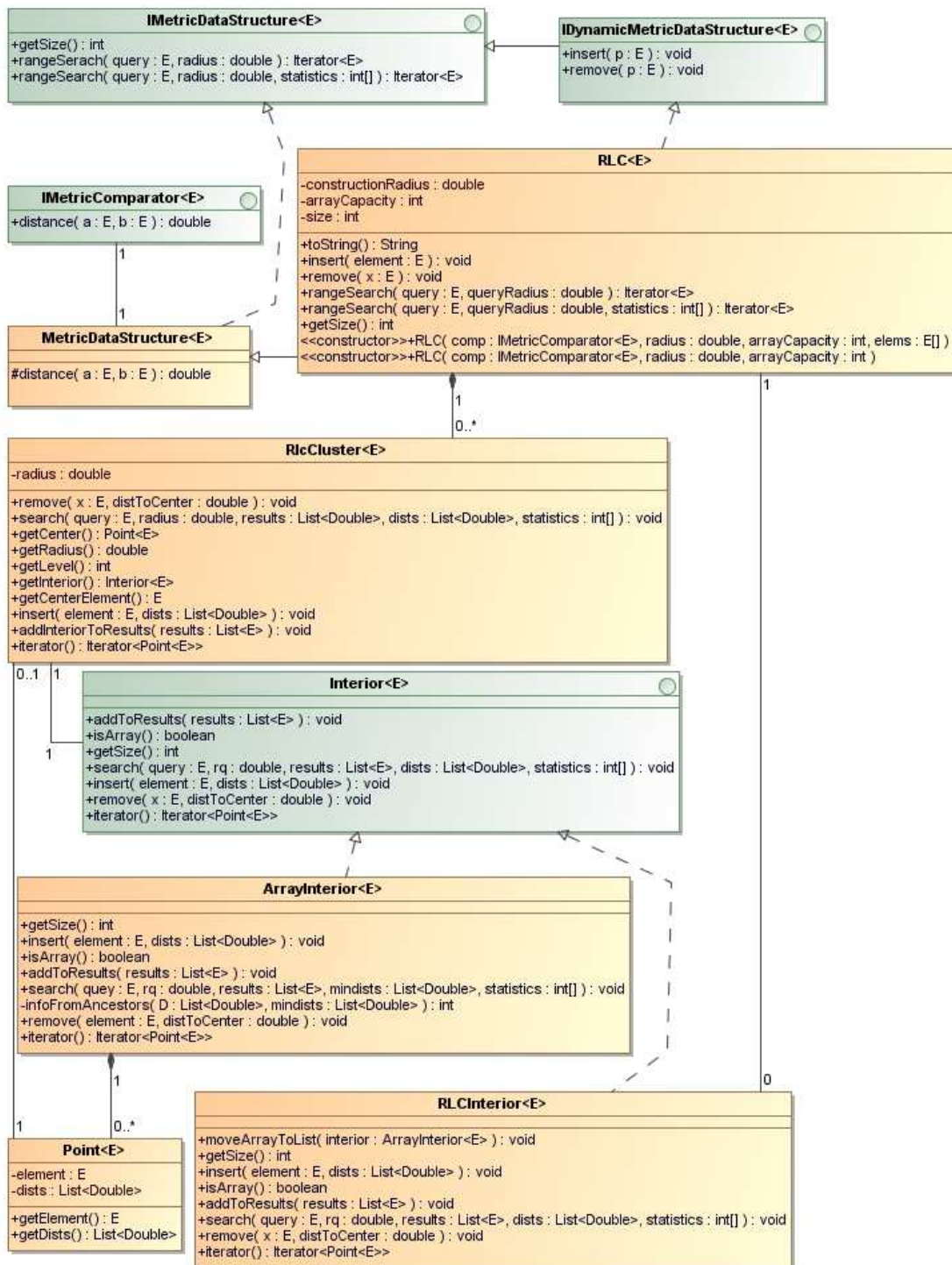


Figura 5.10 – Classe *RLC*

Neste caso foram criados dois construtores, definidos como:

- **public RLC(IMetricComparator<E> comparator, double constructionRadius, int clusterCapacity, E[] elems)**
- **public RLC(IMetricComparator<E> comparator, double constructionRadius, int arrayCapacity)**

A primeira declaração do construtor da *RLC* tem 4 parâmetros que correspondem respectivamente à métrica, ao raio de cada agrupamento, à capacidade de cada vector e ao conjunto dos elementos que serão inseridos na estrutura. A segunda declaração apenas contém os 3 primeiros parâmetros da primeira declaração e é usada para criar uma *RLC* vazia.

## **5.4. Conclusão**

Neste capítulo foi descrito o pacote de estruturas de dados métricas que foi implementado e usado na avaliação das diversas estruturas de dados métricas na pesquisa por alcance das imagens de rosto semelhantes.

O pacote desenvolvido é genérico e, desta forma, poderá ser usado em outros domínios de aplicação. É de salientar que a operação da remoção da *DSAT* e suas variantes não foi implementado.

## **Parte III – Avaliações e protótipo**



## Capítulo 6. Bases de dados utilizadas

Neste capítulo são descritas as bases de dados utilizadas na avaliação dos vários espaços métricos e das estruturas de dados métricas. Todas estas bases de dados foram usadas em trabalhos relacionados com o reconhecimento de rostos [32][34][37]. No entanto, não foi encontrado nenhum trabalho onde tivessem sido utilizadas, juntamente, as 4 bases de dados.

Este capítulo é composto por 5 secções, sendo que as 4 primeiras dizem respeito à caracterização das bases de dados usadas e a última é referente às conclusões.

### 6.1. Base de dados *Faces94*

A base de dados denominada de *Faces94* poderá ser encontrada em [17] onde são apresentadas várias colecções. Neste caso, foi usada a colecção denominada *Faces94* que é composta por imagens de rosto de 152 indivíduos diferentes.

A base de dados é formada por indivíduos de diferentes raças e sexo, com idades maioritariamente compreendidas entre os 18 e os 20 anos, no entanto, também verificamos a existência de alguns indivíduos mais velhos.

Para cada indivíduo existem 20 fotos em que notamos variações de expressão facial consideráveis. No que diz respeito à pose, verificamos pequenas variações na inclinação e da rotação dos rostos.

O fundo de cada imagem é uniforme e é composto por uma cor verde. Em relação às condições de iluminação, notamos que não são apresentadas variações na mesma imagem.

A figura 6.1 retirada de [32] ilustra alguns exemplos de imagens de rosto desta colecção.



Figura 6.1 – Amostras de imagens da base de dados *Faces94*

As imagens de rosto encontram-se no formato *JPEG* de 24 *bits* e têm uma resolução de 180 por 200 *pixels*.

A base de dados *Faces94* foi utilizada em [32], juntamente com a base de dados da *AT&T* [35]. Neste trabalho, foi proposta uma plataforma que combina a decisão de múltiplos classificadores *SVM* – *Support Vector Machines* e esta foi aplicada a um sistema de reconhecimento de rostos. Este sistema apresentou bons resultados nas experiências efectuadas sobre as duas bases de dados. Neste caso verificamos que a taxa de erros na base de dados *Faces94* foi igual a 0% enquanto que na base de dados da *AT&T* a taxa de erros obtida foi de 0.25%.

## 6.2. Base de dados *JAFFE*

A base de dados designada de *JAFFE* poderá ser encontrada em [33] e contém 213 imagens de rosto frontais no formato *TIFF*. Esta base de dados é formada por imagens de rosto associadas a 10 indivíduos japoneses do sexo feminino e cada indivíduo contém entre 20 a 23 fotos. Neste caso verificamos que cada subconjunto de fotos associado à mesma pessoa contém diferentes expressões faciais que denotam vários estados de espírito, tais como felicidade, tristeza, surpresa, raiva, desgosto, medo e expressão neutra. Um exemplo das diferentes variações de expressão poderá ser observada na figura 6.2 retirada de [34].



Figura 6.2 – Exemplo dos vários tipos de expressão da mesma pessoa na *JAFFE*

As várias imagens de rosto encontram-se na escala de cinzento e têm uma dimensão de 256 x 256 *pixels*.

Em relação à iluminação notamos que não são apresentadas variações na mesma imagem.

Esta base de dados foi utilizada em [34] onde foi apresentado um método de extracção de informação associada a expressões faciais em imagens através da aplicação de filtros *Gabor*.

### 6.3. Base de dados *AT&T*

A base de dados da *AT&T* (retirada de [35]) é composta por 400 imagens de rosto referentes a 40 indivíduos e verificamos que contém 10 imagens de rosto por indivíduo.

Nesta base de dados notamos que as várias fotos pertencentes ao mesmo indivíduo contêm diferentes orientações e expressões faciais. Em relação às diferentes expressões faciais podemos encontrar rostos com olhos abertos/fechados, com/sem óculos, entre outras. Um exemplo das diferentes variações de expressão e rotação do rosto poderá ser observada na figura 6.3 retirada de [35].



Figura 6.3 – Exemplo das diferentes variações em imagens associadas ao mesmo indivíduo na *AT&T*

As imagens encontram-se na escala de cinzento, no formato *PGM* e têm a dimensão de 92 x 112 *pixels*.

A base de dados *AT&T* foi utilizada em [37], juntamente com a *Yalefaces*, para comparar um sistema de reconhecimento facial baseado em *SIFT* com os métodos de

*Eigenfaces* [2] e *Fisherfaces* [21]. Neste trabalho o método baseado na técnica *SIFT* foi o que obteve os melhores resultados nas duas bases de dados. Em relação às duas bases de dados a proporção de resultados verdadeiros nos diferentes métodos foi sempre superior com a base de dados *AT&T*.

#### 6.4. Base de dados *Yalefaces*

A base de dados de Yale (retirada de [36]) contém 165 imagens de rosto frontais correspondentes a 14 indivíduos do sexo masculino e 1 do sexo feminino.

Cada indivíduo tem associado 11 imagens onde notamos diferentes variações de expressão e de iluminação. Desta forma, verificamos que cada subconjunto de imagens associado ao mesmo indivíduo contém as seguintes expressões ou configurações: normal, com óculos, sem óculos, feliz, triste, com ar de sono, espantado, a piscar os olhos, ou com a luz a incidir do lado esquerdo, direito ou no centro. Alguns exemplos destas variações estão ilustrados na figura 6.4 retirada de [36].



Figura 6.4 – Exemplo das diferentes variações em imagens associadas ao mesmo indivíduo na *Yalefaces*

As imagens encontram-se na escala de cinzento, no formato *GIF* e têm uma dimensão de 320 x 243 pixéis.

A base de dados *Yalefaces* foi utilizada em [37] para comparar um método baseado na técnica *SIFT* com os métodos de *Eigenfaces*[2] e *Fisherfaces* [21]. O primeiro foi o que obteve os melhores resultados.

#### 6.5. Conclusão

Neste capítulo foram descritas as bases de dados que foram escolhidas para realizar

a avaliação dos espaços métricos e da pesquisa por alcance nas estruturas de dados métricas.

Todas as bases de dados utilizadas têm imagens de rosto com o mesmo tamanho, isto é, numa mesma base de dados todas as imagens têm igual tamanho. A escolha de bases de dados com estas características deve-se ao facto de ter sido utilizado o método de *Eigenfaces* para a representação das imagens.

Em relação ao fundo de cada imagem verificamos, que na maioria das bases de dados apresentadas, as imagens de rosto foram tiradas em ambientes controlados resultando em imagens com fundo uniforme.

A iluminação é outro aspecto importante que pode influenciar o desempenho dos algoritmos de reconhecimento e neste conjunto de bases de dados verificamos que a *Yalefaces* é a base de dados com mais variações das condições de iluminação.

No que diz respeito à rotação dos rostos verificamos que a base de dados da *AT&T* é a que apresenta maiores variações. A base de dados *JAFFE* é a que apresenta maiores variações de expressão facial.

Em síntese verificamos que as bases de dados apresentadas têm diferentes características e permitem testar vários factores como variação da rotação do rosto, diferentes condições de iluminação, variações de expressão facial, entre outros aspectos. No entanto, verificamos que as imagens de rosto das várias bases de dados apresentadas foram tiradas em ambientes controlados, o que nos leva a argumentar que são representativas de aplicações como registos criminais, registos de passaportes, registos de cartas de condução, entre outras.

Outro aspecto relevante está relacionado com a dimensão de cada base de dados. A tabela 6.1 sintetiza o número total de indivíduos e de imagens de rosto para cada base de dados. Como se pode verificar, a *Faces94* é a base de dados com o maior número de imagens.

Base de dados	Nº total de indivíduos	Nº total de imagens de rosto
<i>Faces94</i>	152	3040
<i>JAFFE</i>	10	213
<i>At&amp;t</i>	40	400
<i>Yalefaces</i>	15	165

Tabela 6.1 – Número total de indivíduos e de imagens de rosto nas diferentes bases de dados



## Capítulo 7. Espaços métricos

Relembrando o conceito de espaço métrico como sendo  $(U,d)$  tal que  $U$  é o conjunto de elementos e  $d$  uma métrica. Nesta dissertação, cada elemento do conjunto  $U$  diz respeito a uma representação duma imagem de rosto, com o método de *Eigenfaces*, e  $d$  é uma métrica que pode ser: a distância Euclidiana, a distância de *Manhattan*, *Mahalanobis – L1* e *Mahalanobis – L2*.

De modo a poder avaliar a aplicabilidade das estruturas de dados métricas no domínio das imagens de rosto, foi realizada a implementação destes 4 espaços métricos e foi feita uma avaliação dos mesmos nas 4 bases de dados descritas no capítulo 6.

Assim sendo, este capítulo está estruturado em 4 secções. Na secção 1 será apresentada a descrição da implementação da representação das imagens de rosto e das respectivas métricas. Na secção 2 será apresentado o método utilizado para avaliar as métricas implementadas. Na secção 3 será descrita a avaliação realizada e os seus resultados e, por último, na secção 4 será apresentada a conclusão referente aos espaços métricos usados nesta dissertação.

### 7.1. Implementação

Nesta secção será apresentada a descrição da implementação da representação de imagens de rosto e das respectivas métricas, o qual foi realizado com a linguagem de programação JAVA.

#### 7.1.1. Representação de imagens

Para a representação de imagens de rosto foi utilizado o método designado de *Eigenfaces*, o qual foi implementado com base no código disponível em [16].

A implementação do método de *Eingefaces* encontra-se na classe *Eigenfaces*. Esta classe é responsável por criar o conjunto de vectores próprios e valores próprios com base no conjunto de imagens de treino associado à base de dados. É com base neste conjunto de vectores próprios que é criada uma representação reduzida para cada imagem de rosto.

A classe *FaceImage* corresponde à representação reduzida de uma imagem de rosto, depois de aplicado o método de *Eigenfaces*. Logo, cada imagem de rosto é uma instância da classe *FaceImage*, a qual contém como variáveis de instância um vector de pesos com tamanho igual ao número de vectores próprios usado, o vector de valores próprios e o nome do ficheiro que contém a imagem de rosto.

A classe *FacesRepresentation* é a responsável por carregar a base de dados e criar as respectivas representações das imagens de rosto. O construtor desta classe recebe como parâmetro uma instância da classe *DB* (que contém a informação relativa à base de dados) e tem como função ler do disco o conjunto de instâncias da classe *FaceImage* associado à base de dados. No caso destas não existirem, é necessário recriar o conjunto utilizando para tal a classe *Eigenfaces*.

O conjunto de instâncias da classe *FaceImage* é armazenado num vector na classe *FacesRepresentation*.

Pelo facto de ter sido usado o método de *Eigenfaces* para a representação das imagens de rosto é necessário definir para cada base de dados um conjunto de imagens de rosto de treino. Nesta experiência, o conjunto de imagens de treino usado para cada base de dados contém 25 imagens, o que implica que cada imagem de rosto é representada por um vector com dimensão máxima menor ou igual que 25. Neste caso apenas são usados os vectores próprios cujo respectivo valor próprio é maior que 0.

No que diz respeito ao conjunto de treino não foi realizada nenhuma análise em relação ao tamanho deste e ao número de vectores próprios usados pois esses aspectos estão fora do âmbito desta dissertação.

### **7.1.2. Métricas**

As métricas implementadas foram: Euclidiana, *Manhattan*, *Mahalanobis – L1* e *Mahalanobis – L2*. Desta forma, foram criadas as seguintes classes que implementam a interface *IMetricComparator<FaceImage>*:

- **EuclideanComparatorFaces:** Corresponde à implementação da distância Euclidiana entre duas imagens de rosto. Esta métrica apenas utiliza os respectivos vectores de pesos para medir a similaridade entre dois rostos.
- **CityBlockComparatorFaces:** Implementa a distância de *Manhattan* entre duas imagens de rosto. Neste caso, esta métrica utiliza apenas os respectivos vectores de pesos para medir a similaridade entre dois rostos.
- **MahalanobisL1ComparatorFaces:** Representa a implementação da métrica *Mahalanobis – L1* entre duas imagens de rosto. Neste caso, além dos respectivos vectores de pesos que descrevem a imagem, esta métrica tem em conta o vector de valores próprios.
- **MahalanobisL2ComparatorFaces:** Corresponde à implementação da métrica *Mahalanobis – L2* entre duas imagens de rosto. Esta métrica utiliza os respectivos vectores de pesos e o vector de valores próprios no cálculo da distância entre as duas imagens de rosto.

A figura 7.1 apresenta as classes desenvolvidas para as métricas.

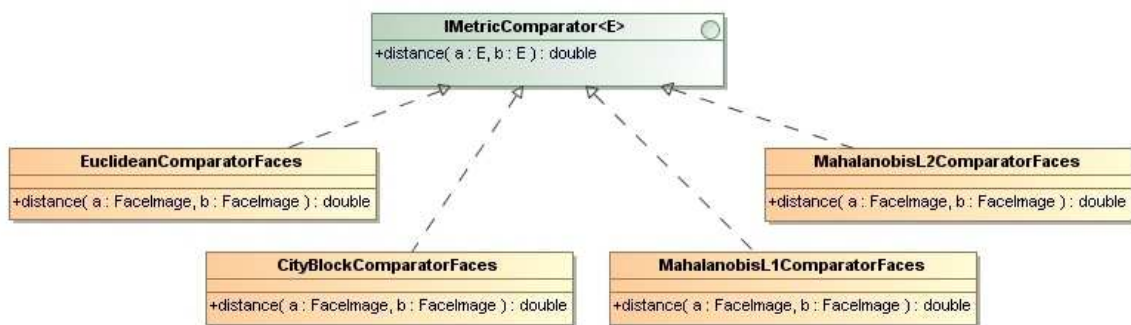


Figura 7.1 – Diagrama de classes das várias métricas

## 7.2. Método de avaliação

Existem várias formas de avaliar o comportamento das várias distâncias, no entanto, todas elas necessitam de uma colecção de objectos e de um conjunto de objectos de pesquisa. Neste caso, para a avaliação dos diferentes espaços métricos foram utilizadas

as bases de dados *Faces94*, *JAFFE*, *AT&T* e *Yalesfaces* descritas anteriormente no capítulo 6.

Os conceitos de elementos relevantes e irrelevantes foram usados na avaliação das várias métricas, para classificar os resultados provenientes das várias consultas. Desta forma, para uma determinada imagem de consulta  $q$ , uma imagem de rosto  $x$  é considerada como sendo relevante se, e apenas se,  $q$  e  $x$  estiverem associados ao mesmo indivíduo. Assim sendo, neste caso a função  $relevante(q,x)$  é verdadeira. No caso de  $q$  e  $x$  não pertencerem ao mesmo indivíduo,  $relevante(q,x)$  é falsa.

Considerando os conceitos de relevantes e irrelevantes a avaliação das várias métricas é baseada nas seguintes medidas:

- **Taxa de Verdadeiros Positivos:** Corresponde à fracção dos elementos relevantes que foram correctamente classificados, ou seja, é a relação entre o número de elementos classificados correctamente como positivos e o número total de elementos positivos existentes. Neste caso, os elementos positivos são todos os elementos relevantes, para a pesquisa em questão, que existem na base de dados.
- **Taxa de Falsos Positivos:** Corresponde ao número de elementos resultantes da pesquisa que são irrelevantes em relação ao número total de elementos irrelevantes existentes na base de dados, ou seja, o número total de elementos que foram incorrectamente classificados como positivos em relação ao número total de negativos. Neste caso, os elementos negativos são todos os elementos irrelevantes que existem na base de dados para a pesquisa em questão.

Seja  $B$  uma base de dados,  $q$  um elemento de pesquisa e  $X$  o conjunto de resultados obtidos da pesquisa tal que  $X \subseteq B$ . A taxa de verdadeiros positivos pode ser vista como a percentagem do número total de elementos relevantes que é obtida por uma pesquisa e é definida da seguinte forma:

$$\text{Taxa de Verdadeiros Positivos} = \frac{|\{x \in X \wedge relevante(q,x)\}|}{|\{x \in B \wedge relevante(q,x)\}|}$$

A taxa de falsos positivos pode ser vista como a percentagem do número total de elementos irrelevantes que é obtido por uma pesquisa e é definido como:

$$\text{Taxa de Falsos Positivos} = \frac{|\{x \in X \wedge \neg \text{relevante}(q, x)\}|}{|B| - |\{x \in B \wedge \text{relevante}(q, x)\}|}$$

Para relacionar a taxa de verdadeiros positivos com a taxa de falsos positivos usaram-se os gráficos *ROC – Receiver Operating Characteristic*.

A análise *ROC* é uma técnica que permite visualizar e avaliar classificadores baseados nas suas performances. Os gráficos *ROC* são bidimensionais, sendo a taxa dos verdadeiros positivos representada no eixo dos YY enquanto que a taxa dos falsos positivos é representada no eixo dos XX. Estes gráficos representam a relação entre os benefícios (taxa de verdadeiros positivos) e os custos (taxa de falsos positivos).

De uma forma geral, um determinado ponto no espaço *ROC*, é melhor que outro se estiver mais a noroeste, ou seja, se a taxa de verdadeiros positivos é maior e/ou a taxa de falsos positivos é menor. Desta forma, o ponto (0,1) no espaço *ROC* representa uma classificação perfeita.

### 7.3. Avaliação realizada e resultados

A avaliação de cada métrica foi realizada com base nas taxas de verdadeiros positivos e de falsos positivos resultantes de um conjunto de pesquisas por semelhança realizadas nas diferentes bases de dados. Desta forma, em cada base de dados de imagens de rosto foram seleccionados aleatoriamente vários elementos para serem usados como objectos de consulta. Para cada elemento de consulta, foi realizada uma pesquisa dos  $k$  vizinhos mais próximos, onde  $k$  corresponde ao número de imagens de rosto associadas ao elemento de consulta que existiam na base de dados. Para cada consulta calculou-se a taxa de verdadeiros positivos e a taxa de falsos negativos e a avaliação foi realizada com base na média de cada taxa referente às várias consultas efectuadas.

De seguida serão descritos o conjunto de pesquisas realizadas e os resultados obtidos.

### 7.3.1. Pesquisas realizadas

Para cada base de dados foi criado um conjunto de elementos de pesquisa formado por diferentes elementos desta. Neste caso, cada conjunto de pesquisa é composto por várias imagens de rosto escolhidas aleatoriamente correspondentes a 25% das imagens associadas a cada indivíduo da base de dados. Assim, verificamos que o conjunto de elementos de pesquisa contém, também, para cada indivíduo, uma percentagem equivalente de imagens associadas, como já ocorria. Formalmente, o conjunto de imagens de rosto de pesquisa  $Q$  associado a uma base de dados  $B$  é definido como:

$$Q = \sum_{i=1}^n A_i$$

Nesta fórmula,  $n$  é o número total de indivíduos pertencentes a  $B$ , e  $A_i$  corresponde ao subconjunto de imagens de rosto escolhidas aleatoriamente pertencentes a  $B$  e associadas ao indivíduo  $i$ . A cardinalidade do conjunto  $A_i$  é igual a 25% do número total de imagens associadas ao indivíduo  $i$ .

A tabela 7.1 sintetiza o número total de indivíduos e de imagens de rosto seleccionadas para cada base de dados.

Base de dados	Nº total de indivíduos	Nº de imagens seleccionadas aleatoriamente
<i>Faces94</i>	152	760
<i>JAFFE</i>	10	50
<i>At&amp;t</i>	40	80
<i>Yalefaces</i>	15	30

Tabela 7.1 – Nº de imagens de rosto seleccionadas em cada base de dados

Seja  $B$  uma base de dados e  $Q$  o conjuntos de objectos de pesquisa ( $Q \subseteq B$ ) e  $k$  o número de imagens de face existentes em  $B$  associadas à imagem de rosto de consulta  $q$ . Para cada imagem de consulta  $q \in Q$ , foi realizada uma pesquisa dos  $k$  vizinhos mais próximos e foi calculada a taxa de verdadeiros positivos e de falsos positivos, do conjunto solução  $X$ . As fórmulas utilizadas foram:

$$\text{Taxa de Verdadeiros Positivos} = \frac{|\{x \in X \wedge \text{relevante}(q, x)\}|}{k}$$

$$\text{Taxa de Falsos Positivos} = \frac{|\{x \in X \wedge \neg \text{relevante}(q, x)\}|}{|B| - k}$$

Neste caso, para cada pesquisa temos garantidamente  $k$  imagens de rosto relevantes e  $|B| - K$  imagens não relevantes.

### 7.3.2. Resultados

Nas figuras 7.2, 7.3, 7.4 e 7.5 estão representados os gráficos *ROC* associados a esta avaliação para cada métrica, nas diferentes bases de dados. Os resultados apresentados são a média das taxas de verdadeiros positivos e taxas de falsos positivos obtidos do conjunto de pesquisa de cada base de dados.

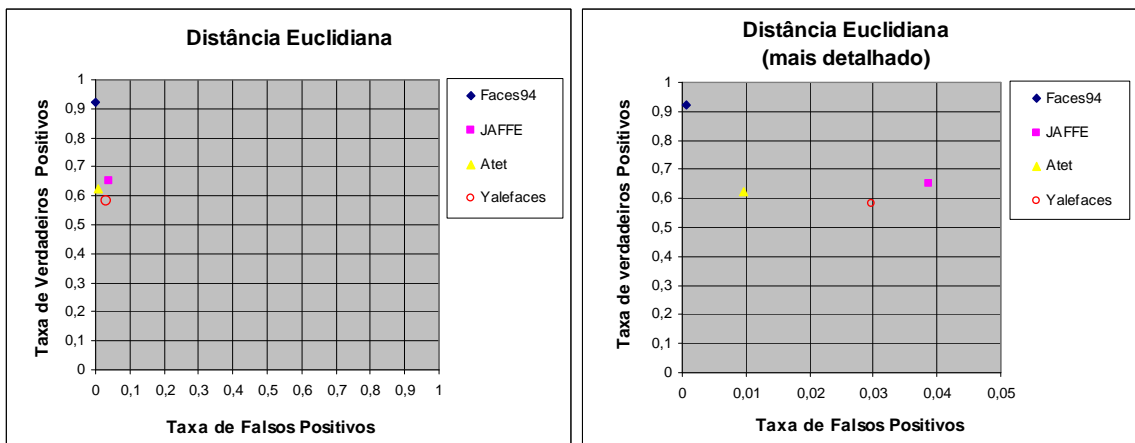


Figura 7.2 – Avaliação da distância Euclidiana

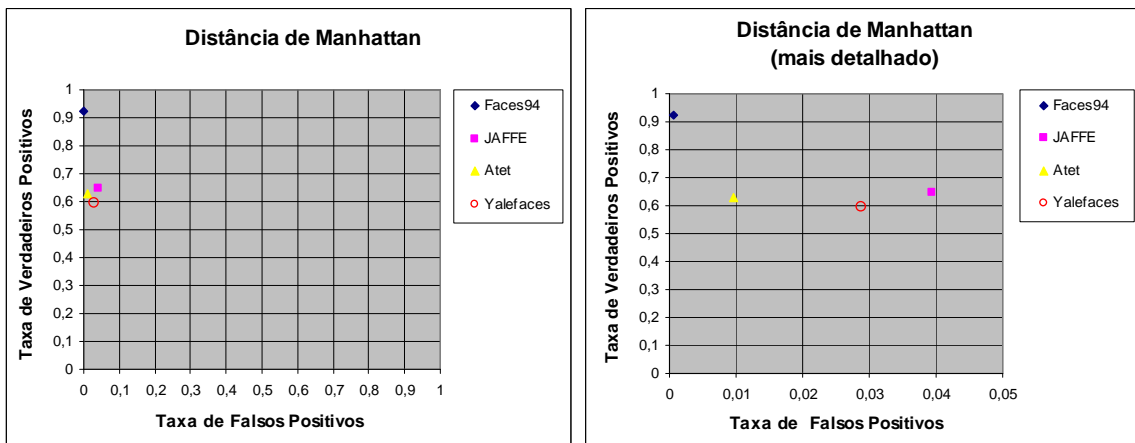


Figura 7.3 – Avaliação da distância de *Manhattan*

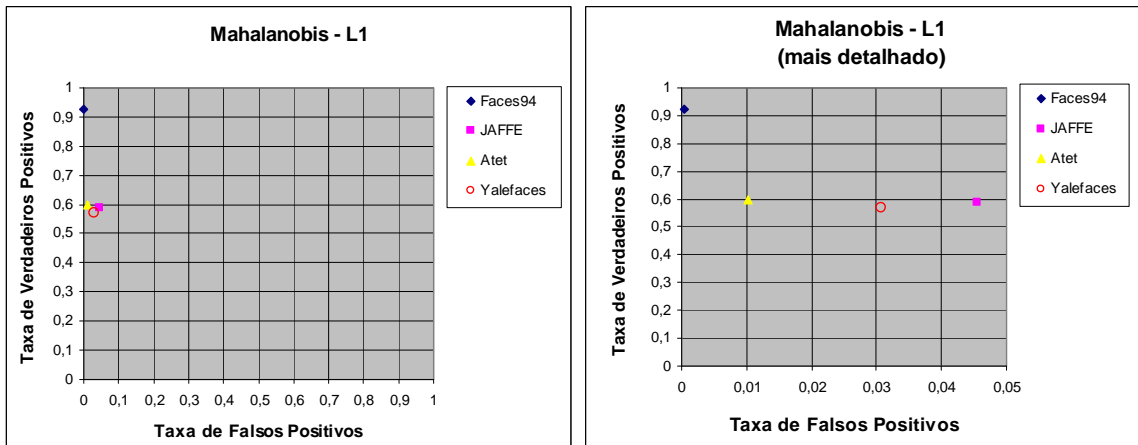


Figura 7.4 – Avaliação da métrica *Mahalanobis – L1*

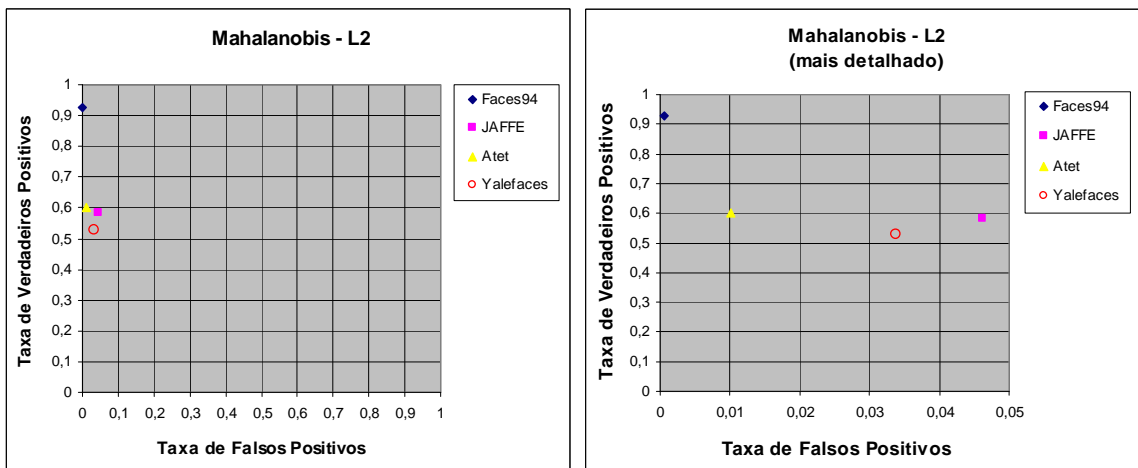


Figura 7.5 – Avaliação da métrica *Mahalanobis – L2*

Como podemos verificar nestes gráficos, as quatro métricas tiveram desempenho muito semelhantes nas diferentes bases de dados. Neste caso, verificamos que todas as métricas obtiveram simultaneamente o maior valor da taxa de verdadeiros positivos e o menor valor da taxa de falsos positivos na base de dados *Faces94*. Neste caso temos uma classificação quase perfeita das várias métricas, muito próxima do ponto (0,1), pois a taxa de verdadeiros positivos foi, em todos os casos, superior a 0,92 e a taxa de falsos positivos foi sempre inferior a 0,0006.

Outro aspecto que verificamos é que, em todas as métricas, a base de dados *JAFFE* foi a que obteve a maior taxa de falsos positivos, ainda que com valores abaixo de 0,047.

A figura 7.6 apresenta a média das taxas de verdadeiros positivos e falsos positivos de todas as bases de dados, para cada uma das métricas.

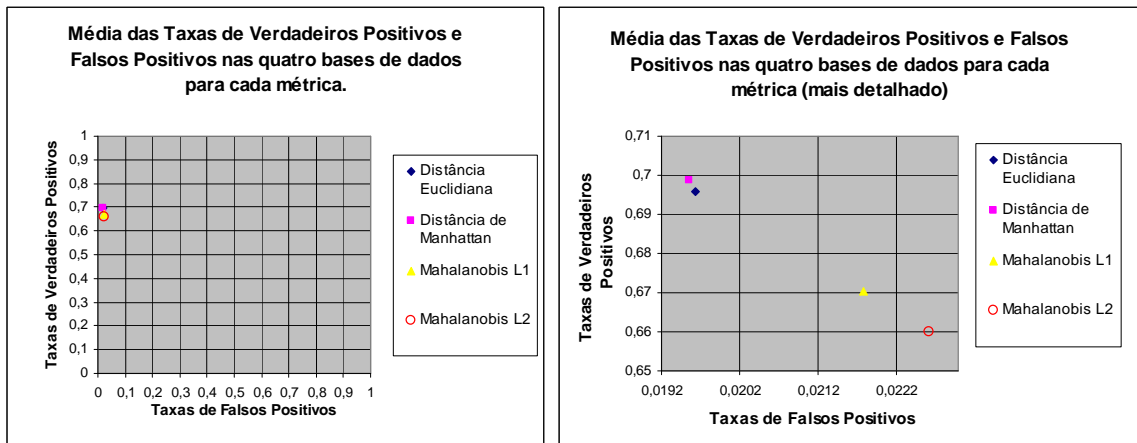


Figura 7.6 – Gráfico ROC com a média das taxas de verdadeiros e falsos positivos das quatro bases de dados para cada métrica.

Pela análise do gráfico da figura 7.6, verificamos a média da taxa de verdadeiros positivos de qualquer uma das métricas nas 4 bases dados varia entre 0,66 e 0,7. Em relação à média da taxa de falsos positivos nas 4 bases de dados verificamos que varia entre 0,0192 e 0,0232.

Outro aspecto que é possível verificar é que, apesar das várias métricas apresentarem uma performance muito semelhante, a distância de *Manhattan* foi a métrica que obteve o ponto mais próximo do ponto óptimo nas 4 bases de dados, seguindo-se a *Euclidiana*, a *Mahalanobis – L1* e, por último, a *Mahalanobis – L2*.

## 7.4. Conclusão

Neste capítulo foi apresentado a implementação dos espaços métricos, no que diz respeito à representação das imagens de rosto e das métricas, e a avaliação destas métricas nas 4 bases de dados usadas. As métricas utilizadas para a definição dos vários espaços métricos foram a distância *Euclidiana*, distância de *Manhattan*, *Mahalanobis – L1* e *Mahalanobis – L2*.

A pesquisa por similaridade com estas métricas apresenta bons resultados nestas bases de dados, pois nos resultados obtidos das diversas pesquisas estão a maioria dos elementos relevantes e a maioria dos elementos irrelevantes não aparecem. Neste caso, a taxa de verdadeiros positivos varia entre 0,66 e 0,7 e a taxa de falsos positivos varia entre 0,0192 e 0,0232.

As 4 métricas apresentaram performances muito semelhantes, no entanto, uma

análise mais detalhada mostra que a distância de *Manhattan* foi a melhor, seguindo-se a distância Euclidiana. Neste caso, a métrica *Mahalanobis* – *L2* foi a que obteve, simultaneamente, a menor taxa de verdadeiros positivos e a maior taxa de falsos positivos.

Por esta avaliação podemos concluir que é possível definir espaços métricos no domínio das imagens de rosto nestas 4 bases de dados, os quais têm resultados “bons” na pesquisa por semelhança.

## Capítulo 8. Pesquisa por alcance com estruturas de dados métricas

De modo a poder avaliar a eficiência da pesquisa por alcance com estruturas de dados métricas, no domínio das imagens de rosto, foram realizadas um conjunto de pesquisas nos vários espaços métricos implementados. Neste caso, foram utilizados os 4 espaços métricos apresentados, nas 4 bases de dados (*Faces94*, *JAFFE*, *AT&T* e *Yalefaces*) e foi avaliada a eficiência da pesquisa por alcance nas estruturas de dados métricas: *LAESA*, *VPTree*, *DSAT*, *HDSAT1*, *HDSAT2*, *GNAT*, *LC*, *RLC* e *GNAT*.

Em relação a esta avaliação, existem vários aspectos que devem ser tomados em consideração. Estes aspectos são:

- Os dados a processar;
- A parametrização das estruturas de dados métricas.

No que diz respeito aos dados a processar, é necessário ter em atenção três aspectos: (1) a base de dados a utilizar, (2) o conjunto de consulta deve ser representativo do conjunto de elementos da base de dados e (3) os raios de pesquisa devem produzir resultados relevantes na aplicação em causa.

A maioria das estruturas de dados necessita de ser parametrizada pois o seu comportamento depende dos parâmetros escolhidos. Estes parâmetros dependem seguramente da caracterização do espaço métrico no que diz respeito às distâncias entre os vários elementos<sup>1</sup>. Uma vez escolhidos os dados a processar e as parametrizações a realizar, foram realizadas as várias pesquisas por alcance e foram analisados os resultados de acordo com o número de distâncias calculadas em cada pesquisa.

(1) Existe uma tese de mestrado em curso que pretende estudar esta problemática na *RLC*. A tese intitula-se “Modelação da Estrutura de dados métrica *RLC*” e é orientada pela professora Margarida Mamede.

## **8.1. Dados a processar**

Esta secção apresenta os dados que serão processados (base de dados, conjunto de elementos de consulta e os raios de pesquisa), ou seja, é definido o conjunto de consulta utilizado nas pesquisas por alcance.

### **8.1.1. Base de dados e conjuntos de consulta**

Para a realização da avaliação foram gerados 4 ficheiros para cada base de dados. O mais pequeno corresponde ao conjunto de objectos de pesquisa e é formado por elementos aleatórios da base de dados. De modo a manter o mesmo número de indivíduos diferentes neste conjunto, foram escolhidos 25% das imagens de face associadas a cada indivíduo. Este conjunto de objectos foi o mesmo que foi utilizado na avaliação das diferentes métricas, como podemos verificar em 7.3.1.1.

Os outros 3 ficheiros correspondem a permutações aleatórias da base de dados e foram usados na construção das diversas estruturas de dados métricas. Este aspecto está relacionado com o facto da organização de cada estrutura de dados métrica depender da ordem segundo a qual os objectos ocorrem como parâmetro de entrada para o algoritmo de construção.

### **8.1.2. Raios de pesquisa**

A escolha do raio de pesquisa é um factor importante a ter em conta. De modo a obter um conjunto de resultados relevantes, para as pesquisas em questão, seria aconselhável poder estimar quais os raios mais pertinentes para cada espaço métrico e base de dados, até porque estes serão os raios que os utilizadores usarão com mais regularidade. Tendo como base este facto e a necessidade de parametrizar as estruturas de dados métricas de modo a obter bons resultados nas pesquisas, foi realizada uma análise para a selecção do raio de pesquisa. Esta avaliação envolveu diferentes pesquisas por alcance com diferentes raios. Para cada espaço métrico numa determinada base de dados foram testados 1000 raios no intervalo de 0 a  $M$ , onde  $M$  corresponde à distância máxima que existe entre os vários elementos. Desta forma, para cada raio

foram realizadas várias pesquisas por alcance com o respectivo conjunto de consulta. Note-se que nestas pesquisas só se utilizou uma das 3 permutações geradas, pois só estávamos interessados nos resultados e não no desempenho das estruturas de dados métricas.

Com base nos resultados de cada pesquisa, foi calculada a taxa de verdadeiros positivos, a taxa de falsos positivos e o número de elementos retornados da pesquisa. Estas taxas foram calculadas com os mesmos critérios que foram apresentados em 7.2. Para cada raio foi obtida a média da taxa de verdadeiros positivos, a média da taxa de falsos positivos e a média do número de resultados obtidos das várias pesquisas por alcance realizadas com esse raio.

Tendo como objectivo maximizar a taxa de verdadeiros positivos, minimizar a taxa de falsos positivos e obter um número de respostas razoáveis na aplicação em causa, foram identificados, para cada base de dados num dado espaço métrico, os raios candidatos. Estes raios tinham que satisfazer as seguintes condições:

- A média da taxa de falsos positivos é inferior a 0.03.
- A média do número de elementos do conjunto de resultados das várias pesquisas é não superior a 130% da média do número de imagens associadas a cada um dos indivíduos da base de dados. Por exemplo, numa base de dados que contenha em média 10 imagens de rosto por indivíduo, a média do número de resultados das várias pesquisas com um determinado raio deverá ser não superior a 13 imagens. Desta forma, pretende-se que o número máximo de elementos retornados de uma pesquisa não seja excessivo.
- Não existe nenhum raio que tenha simultaneamente uma média da taxa de verdadeiros positivos superior ou igual e uma taxa de falsos positivos inferior ou igual.

Do conjunto de raios candidatos foi seleccionado o raio cuja média da taxa de verdadeiros positivos e falsos positivos está mais próxima do ponto óptimo (que corresponde a uma taxa de verdadeiros positivos igual a 1 e uma taxa de falsos positivos igual a 0).

A tabela 8.1 apresenta o raio que foi seleccionado para cada base de dados num dado espaço métrico.

Métrica	Raio	Taxa de verdadeiros positivos	Taxa de falsos positivos	Média do nº de elementos retornados
<b>Faces94</b>				
<b>Euclidiana</b>	<b>3338</b>	0,9032	0,0025	25,74
<i>Manhattan</i>	<b>12347</b>	0,9014	0,0026	25,85
<i>Mahalanobis-L1</i>	<b>7,96</b>	0,8967	0,0026	25,76
<i>Mahalanobis-L2</i>	<b>2,04</b>	0,8992	0,0026	25,78
<b>JAFFE</b>				
<b>Euclidiana</b>	<b>4649</b>	0,4975	0,0297	16,34
<i>Manhattan</i>	<b>15560</b>	0,4794	0,0294	15,9
<i>Mahalanobis-L1</i>	<b>10,23</b>	0,458	0,0287	15,32
<i>Mahalanobis-L2</i>	<b>2,74</b>	0,4676	0,0295	15,7
<b>AT&amp;T</b>				
<b>Euclidiana</b>	<b>1507</b>	0,6638	0,016	12,89
<i>Manhattan</i>	<b>5447</b>	0,6525	0,0159	12,74
<i>Mahalanobis-L1</i>	<b>10,46</b>	0,6375	0,0169	12,96
<i>Mahalanobis-L2</i>	<b>2,71</b>	0,6388	0,0167	12,9
<b>Yalefaces</b>				
<b>Euclidiana</b>	<b>9400</b>	0,5242	0,0273	9,97
<i>Manhattan</i>	<b>33767</b>	0,5363	0,0296	10,47
<i>Mahalanobis-L1</i>	<b>14,95</b>	0,5091	0,0284	9,97
<i>Mahalanobis-L2</i>	<b>4,02</b>	0,4576	0,029	9,5

Tabela 8.1 – Raios de pesquisa utilizados na parametrização das estruturas e respectivas médias das taxas de verdadeiros e falsos positivos e do nº de resultados obtidos das várias pesquisas

Estes raios seleccionados serão utilizados na parametrização das estruturas de dados métricas e, com base nestes valores, serão avaliadas as pesquisas por alcance nas estruturas de dados métricas. Para além disso, estes serão os raios que aparecerão por defeito cada vez que é seleccionada uma base de dados e uma métrica no protótipo implementado (ver capítulo 9).

## 8.2. Parametrização das estruturas de dados métricas

Esta secção descreve as parametrizações usadas nas diversas estruturas de dados métricas. De modo a saber a parametrização que melhor desempenho obtinha na pesquisa por alcance nas diversas bases de dados (menor número de computações da métrica), foi realizado um estudo que tinha como objectivo escolher uma parametrização para realizar a avaliação final que fosse “justa” para todas as estruturas de dados métricas. Para mais detalhe sobre este estudo consultar anexo A 3.

Neste estudo, é de salientar que a parametrização das diversas estruturas de dados métricas dependia da base de dados sobre o espaço métrico. Como pretendíamos usar 4 bases de dados sobre 4 espaços métricos, existem 16 casos experimentais por estrutura de dados. Note-se que a estrutura de dados *VPTree* é a única que não tem parametrizações.

De seguida serão apresentadas as parametrizações usadas, para cada estrutura de dados métrica, as quais resultaram da análise dos gráficos apresentados no anexo A 3.

### 8.2.1. LAESA

Na *LAESA* é possível parametrizar o número de elementos que será usado como *pivots*. A tabela 8.2 apresenta, para cada espaço métrico nas diferentes bases de dados, o número de *pivots* usados na avaliação final.

	<b>Distância Eucliana</b>	<b>Distância de Manhattan</b>	<b><i>Mahalanobis - L1</i></b>	<b><i>Mahalanobis - L2</i></b>
<i>Faces94</i>	46	67	55	67
<i>JAFFE</i>	15	10	27	27
<i>AT&amp;T</i>	30	28	44	59
<i>Yalefaces</i>	11	15	31	29

Tabela 8.2 – Número de *pivots* na *LAESA*

### 8.2.2. DSAT e variantes

Nas estruturas de dados *DSAT*, *HDSAT1* e *HDSAT2* é necessário parametrizar a aridade máxima. A tabela 8.3 indica a configuração da aridade máxima usada na avaliação final.

		<b>Distância Eucliana</b>	<b>Distância de Manhattan</b>	<b>Mahalanobis- L1</b>	<b>Mahalanobis- L2</b>
<i>DSAT</i>	<i>Faces 94</i>	3	6	9	3
	<i>JAFFE</i>	3	6	8	4
	<i>AT&amp;T</i>	7	6	10	8
	<i>Yalefaces</i>	3	8	10	9
<i>HDSAT1</i>	<i>Faces94</i>	6	6	8	8
	<i>JAFFE</i>	4	9	6	4
	<i>AT&amp;T</i>	8	6	11	10
	<i>Yalefaces</i>	8	9	10	12
<i>HDSAT2</i>	<i>Faces94</i>	6	9	9	8
	<i>JAFFE</i>	7	6	6	10
	<i>AT&amp;T</i>	8	13	13	10
	<i>Yalefaces</i>	8	9	11	12

Tabela 8.3 – Aridade máxima na *DSAT*, *HDSAT1* e *HDSAT2*

### 8.2.3. GNAT

Na *GNAT* é necessário definir o número máximo de centros de cada nó. É de salientar que nesta estrutura de dados, para seleccionar a parametrização a usar na avaliação final, foi necessário ter em conta o espaço em memória ocupado pela estrutura de dados (ver anexo A 3.2), no entanto como se poderá ver na avaliação final esta não foi injustamente prejudicada. Desta forma, para o teste final, as estruturas *GNAT* foram todas parametrizadas com um número de centros máximo igual a 3. Neste caso, um número de centros máximos igual a 3 é suficiente para competir com as várias estruturas de dados métricas. Em alguns casos, com esta parametrização, a *GNAT* conseguiu ser a segunda melhor estrutura (ver secção 8.4.1).

### 8.2.4. LC

Nesta experiência foi utilizada a *LC* com agrupamentos de raio fixo, logo é necessário parametrizar o raio de construção de cada agrupamento. A tabela 8.4 apresenta os valores do raio do agrupamento usados.

	<b>Distância Eucliana</b>	<b>Distância de Manhattan</b>	<b>Mahalanobis-L1</b>	<b>Mahalanobis-L2</b>
<i>Faces94</i>	2871	9799	5,99	1,65
<i>JAFFE</i>	6637	15862	8,09	2,04
<i>AT&amp;T</i>	1522	5477	7,68	2,72
<i>Yalefaces</i>	10378	26300	7,81	2,01

Tabela 8.4 – Raios de construção da *LC*

### 8.2.5. RLC

A *RLC* tem duas propriedades que podem ser parametrizadas: raio de cada agrupamento e capacidade de cada vector. A tabela 8.5 apresenta a parametrização final escolhida para o raio de construção da *RLC*.

	<b>Distância Eucliana</b>	<b>Distância de Manhattan</b>	<b>Mahalanobis – L1</b>	<b>Mahalanobis – L2</b>
<i>Faces94</i>	2776	10445	6,19	1,58
<i>JAFFE</i>	5790	17486	10,34	2,76
<i>AT&amp;T</i>	1511	5514	10,52	2,72
<i>Yalefaces</i>	10310	35067	15,32	4,06

Tabela 8.5 – Raio de cada agrupamento da *RLC*

A tabela 8.6 apresenta os valores da capacidade do vector usados na *RLC*.

	<b>Distância Eucliana</b>	<b>Distância de Manhattan</b>	<b>Mahalanobis – L1</b>	<b>Mahalanobis – L2</b>
<i>Faces94</i>	18	18	17	17
<i>JAFFE</i>	16	18	10	11
<i>AT&amp;T</i>	6	7	7	5
<i>Yalefaces</i>	7	13	7	5

Tabela 8.6 – Capacidade do vector na *RLC*

### 8.3. Avaliação das estruturas de dados

A avaliação final mostra o comportamento das várias estruturas de dados métricas nas diferentes bases de dados sobre os espaços métricos, nas pesquisas por alcance com vários raios de pesquisa.

Para cada base de dados sobre um espaço métrico, foi realizado o mesmo conjunto de pesquisas sobre as 3 permutações criadas na base de dados (ver secção 8.1.1).

Esta avaliação envolveu a construção de 3 instâncias de estruturas de dados, para cada estrutura de dados métrica usada. Na figura 8.1 pode-se ver este processo na base de dados *Faces94* com a estrutura de dados métrica *RLC*.

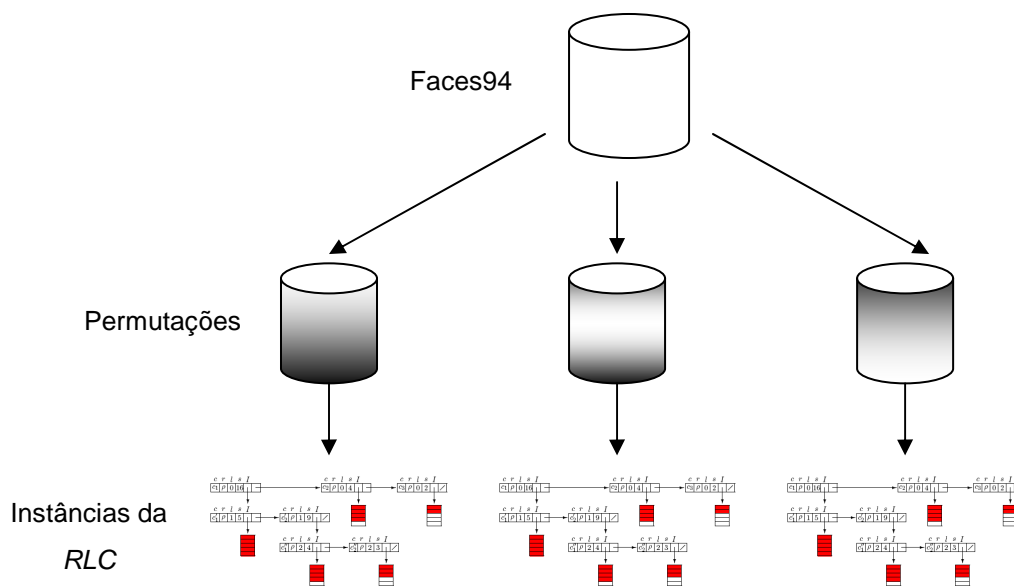


Figura 8.1 – Construção de 3 instâncias da *RLC* com a base de dados *Faces94*

As pesquisas realizadas foram sempre feitas sobre o conjunto de consulta definido em 8.1.1 e com os raios seleccionados em 8.1.2. Ou seja, para cada instância da estrutura de dados e um dado raio de pesquisa foram realizadas  $n$  pesquisas por alcance, sendo  $n$  o número de elementos do conjunto de consulta. Para este conjunto de pesquisas foi calculada a média do número de cálculos da métrica realizada em cada pesquisa. Os resultados apresentados para cada estrutura de dados numa base de dados são a média destes valores para cada instância da estrutura de dados (ou seja, cada permutação da base de dados).

É de salientar que nesta avaliação as estruturas de dados foram parametrizadas de acordo com o que foi apresentado na secção 8.2.

Em relação às estruturas de dados *DSAT* e variantes, apenas foi utilizada a *HDSAT2* nesta avaliação pois, das 3, foi a que obteve sempre os melhores resultados (ver anexo A 3.3).

### 8.3.1. Raios de pesquisa utilizados na avaliação

Para o teste final foram usados 3 raios de pesquisa para cada espaço métrico na respectiva base de dados. Neste caso, os 3 raios de pesquisa correspondem respectivamente a 50%, 100% e 120% do raio seleccionado (ver secção 8.1.2) que também foi utilizado na parametrização das estruturas de dados métricas.

Os 3 raios de pesquisa utilizados para cada métrica, nas diferentes bases de dados estão representados na tabela 8.7.

	Distância Eucliana			Distância de Manhattan			Mahalanobis – L1			Mahalanobis – L2		
	1º raio	2º raio	3º raio	1º raio	2º raio	3º raio	1º raio	2º raio	3º raio	1º raio	2º raio	3º raio
<b>Faces94</b>	1669	3338	4006	6174	12347	14816	3,98	7,96	9,55	1,02	2,04	2,45
<b>JAFFE</b>	2325	4649	5579	7780	15560	18672	5,12	10,23	12,28	1,37	2,74	3,29
<b>AT&amp;T</b>	754	1507	1808	2724	5447	6536	5,23	10,46	12,55	1,36	2,71	3,25
<b>Yalefaces</b>	4700	9400	11280	16884	33767	40520	7,48	14,95	17,94	2,01	4,02	4,82

Tabela 8.7 – Raios de pesquisa utilizados na avaliação das pesquisas por alcance

Na tabela 8.8 é apresentado, para cada raio de pesquisa, a média do nº de elementos obtidos na pesquisa por alcance, dos vários objectos de pesquisa nas diferentes métricas.

	Distância Eucliana			Distância de Manhattan			Mahalanobis – L1			Mahalanobis – L2		
	1º raio	2º raio	3º raio	1º raio	2º raio	3º raio	1º raio	2º raio	3º raio	1º raio	2º raio	3º raio
<b>Faces94</b>	13,29	25,74	56,37	13,04	25,84	60,69	12,42	25,07	63,85	12,43	25,41	65,4
<b>JAFFE</b>	5,38	16,34	23,84	4,78	15,9	25,28	3,92	15,32	27,34	3,74	15,7	28,22
<b>AT&amp;T</b>	2,38	12,89	30,3	2,23	12,73	32,6	1,99	12,96	37,95	2,01	12,98	38,05
<b>Yalefaces</b>	3,27	9,97	15,3	3,4	10,47	17,17	2,47	9,97	26,17	2,1	9,47	27,13

Tabela 8.8 – Média do nº de elementos obtidos com os diferentes raios associados a cada uma das métricas

Verificamos que os diferentes raios que foram seleccionados permitem testar o comportamento das várias estruturas de dados em pesquisas de curto (1º raio), médio (2º raio) ou grande alcance (3º raio).

## 8.4. Resultados da avaliação

Nesta secção serão apresentados os resultados da avaliação final das várias estruturas de dados métricas para as diferentes métricas. Os valores apresentados para cada base de dados dizem respeito à média dos valores obtidos em cada permutação da base de dados.

### 8.4.1. Avaliação das estruturas de dados com a distância Euclidiana

O gráfico da figura 8.2 mostra, para as diferentes bases de dados e diversos raios de pesquisa, a média do número de computações da distância realizadas nas pesquisas por alcance com as diferentes estruturas de dados métricas, tendo como métrica a distância Euclidiana. A tabela 8.9 apresenta a percentagem da média do número de computações efectuadas com as diversas estruturas de dados métricas em relação ao tamanho das várias bases de dados para cada pesquisa por alcance realizada.

	<i>Faces94</i>			<i>JAFFE</i>			<i>AT&amp;T</i>			<i>Yalefaces</i>		
	1º raio	2ºraio	3º raio	1º raio	2º raio	3º raio	1ºraio	2ºraio	3º raio	1ºraio	2ºraio	3ºraio
<i>VPtree</i>	7,43%	20,98%	26,95%	18,99%	39,06%	45,9%	27,87%	60,79%	72,02%	28,93%	53,39%	61,37%
<i>LC</i>	5,97%	15,33%	21,16%	20,16%	36,03%	42,98%	22,71%	46,83%	56,48%	19,93%	40,07%	48,12%
<i>HDSAT2</i>	4,68%	10,67%	14,51%	17,72%	31,11%	37,5%	21,89%	45,61%	56,79%	19,05%	35,19%	43,62%
<i>RLC</i>	5,9%	11,38%	14,07%	17,9%	30,06%	34,46%	25,61%	43,67%	55,74%	19,7%	33,64%	42,51%
<i>GNAT</i>	4,56%	11,07%	15,2%	14,46%	30,04%	37,12%	22,48%	50,82%	62,43%	16,92%	37,91%	47,8%
<i>LAESA</i>	2,94%	14,14%	21,71%	12,95%	24,53%	30,96%	10,42%	32,36%	45,73%	13,76%	31,33%	40,30%

Tabela 8.9 – Percentagem da média do nº de computações efectuadas em relação ao tamanho da base de dados com a distância Euclidiana

É possível observar nestes resultados que, face à pesquisa exaustiva, houve uma redução do número de computações da métrica entre 73,05% a 97,06% na base de dados *Faces94*, entre 54,1% a 87,05% na *JAFFE*, entre 27,89% a 89,58% na *AT&T* e entre 38,63% a 86,24% na *Yalefaces*.

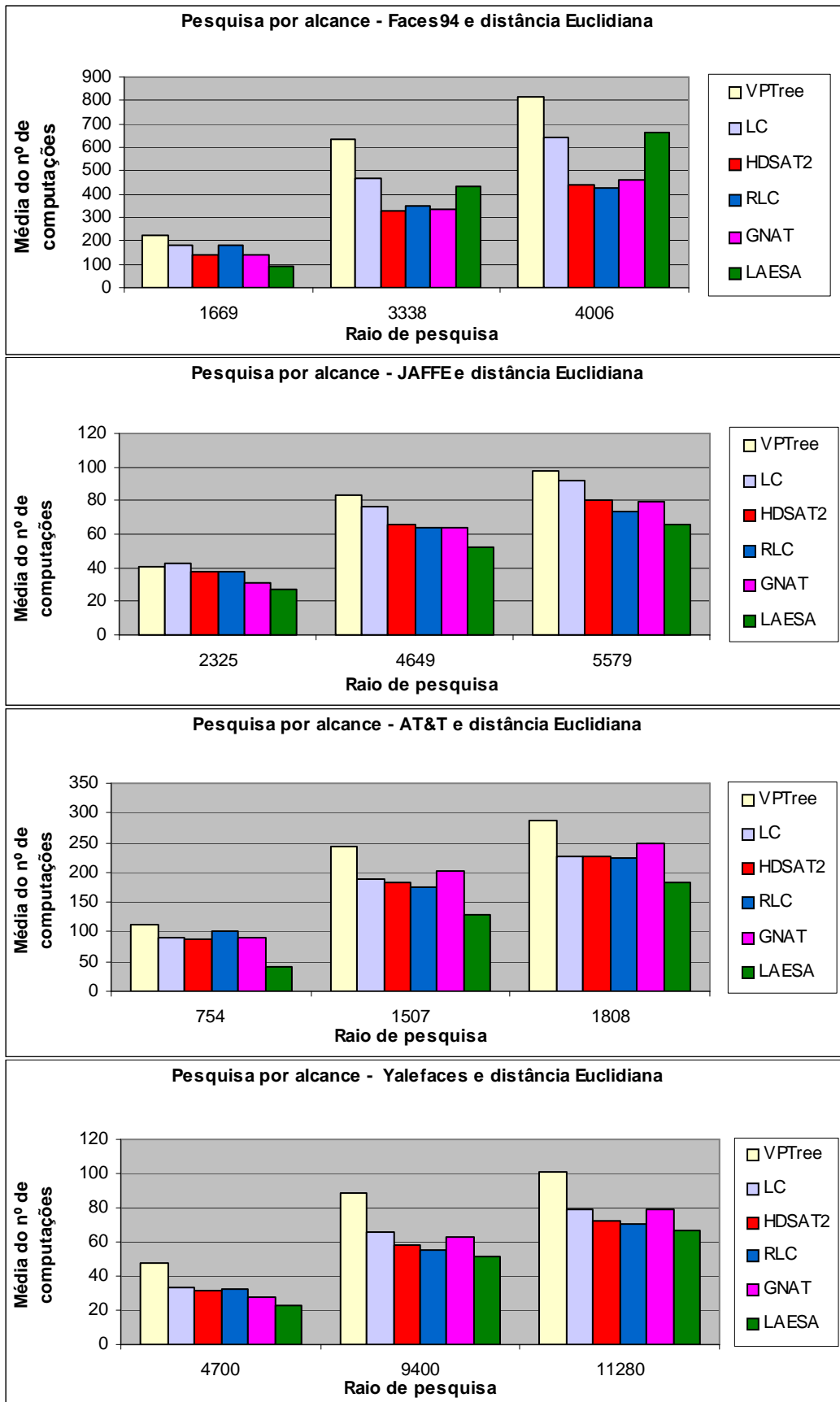


Figura 8.2 – Média do nº de computações das várias estruturas de dados métricas nas diferentes bases de dados e com a distância Euclidiana

Em relação às estruturas de dados métricas, a *LAESA* foi a que obteve os melhores resultados na maioria dos casos (ver cor amarela, na tabela 8.9). Neste caso, verificamos que obteve a menor média do número de computações nas bases de dados *JAFFE*, *AT&T* e *Yalefaces*. Na base de dados *Faces94*, a *LAESA* apenas foi a melhor com o primeiro raio de pesquisa. Nesta base de dados, a *HDSAT2* teve o melhor desempenho com o segundo raio de pesquisa e a *RLC* foi a melhor com o terceiro raio.

Excluindo a *LAESA*, as estruturas *GNAT* e *RLC* foram as estruturas que tiveram o melhor desempenho. Neste caso, verificamos que cada uma destas estruturas foi a segunda melhor em 5 dos 12 casos experimentais, como se pode comprovar na tabela 8.9 (ver cor verde).

Em relação às estruturas de dados dinâmicas (*RLC* e *HDSAT2*), verificamos que a *RLC* foi a que obteve os melhores resultados na maioria dos casos experimentais (7 em 12).

É de salientar que as estruturas de dados métricas com os piores resultados, na maioria dos casos, foram a *LC* e a *VPtree*.

Em relação à degradação do desempenho das estruturas de dados, entre o primeiro e o terceiro raio de pesquisa, verificamos que as que se degradaram mais foram a *VPtree* e a *LAESA* na base de dados *Faces94*, a *VPtree* e a *LC* na *JAFFE*, a *VPtree* e a *GNAT* na *AT&T* e a *VPtree* e *GNAT* na *Yalefaces*. A *RLC* foi a estrutura que apresentou a menor degradação nas diversas bases de dados.

#### **8.4.2. Avaliação das estruturas de dados com a distância de *Manhattan***

O gráfico da figura 8.3 mostra, para as diferentes bases de dados e diversos raios de pesquisa, a média do número de computações da distância realizadas nas pesquisas por alcance com as diferentes estruturas de dados métricas, tendo como métrica a distância de *Manhattan*. A tabela 8.10 apresenta a percentagem da média do nº de computações efectuadas com as diversas estruturas de dados métricas em relação ao tamanho das várias bases de dados nas pesquisas realizadas.

Os resultados apresentados mostram que, face à pesquisa exaustiva, houve uma redução do número de computações da métrica entre 63,56% a 96,81% na base de dados *Faces94*, entre 36,66% a 87,77% na *JAFFE*, entre 13,03% a 90,44% na *AT&T* e entre 19,08% a 84,64% na *Yalefaces*.

	<i>Faces94</i>			<i>JAFFE</i>			<i>AT&amp;T</i>			<i>Yalefaces</i>		
	1º raio	2º raio	3º raio	1º raio	2º raio	3º raio	1º raio	2º raio	3º raio	1º raio	2º raio	3º raio
<i>VPTree</i>	10,64%	28,86%	36,44%	27,86%	53,94%	63,34%	38,53%	78,48%	86,97%	45,14%	72,53%	80,92%
<i>LC</i>	6,69%	18,41%	25,83%	23,78%	40,67%	50,2%	26,67%	56,17%	69,9%	26,59%	48,45%	60,92%
<i>HDSAT2</i>	6,54%	14,54%	19,4%	22,56%	40,46%	48,74%	29,37%	58,64%	71,06%	24,55%	47,25%	58,32%
<i>RLC</i>	5,97%	12,14%	15,59%	20,24%	36,70%	45,69%	27,95%	53,79%	69,1%	22,24%	43,06%	56,54%
<i>GNAT</i>	7,12%	16,08%	21,2%	19,94%	41,37%	51%	33,62%	65,84%	76,68%	23,58%	51,2%	63,44%
<i>LAESA</i>	3,19%	17,08%	28,56%	12,23%	31,73%	43,01%	9,56%	38,28%	55,73%	15,36%	34,61%	46,63%

Tabela 8.10 – Percentagem da média do nº de computações efectuadas em relação ao tamanho da base de dados com a distância de *Manhattan*

Em relação às estruturas de dados métricas, a *LAESA* foi a que obteve a menor média do nº de computações da distância de *Manhattan* em 10 de 12 casos experimentais (ver cor amarela na tabela 8.10). Nos outros dois casos, foi a *RLC* que obteve os melhores resultados.

Excluindo a *LAESA*, a *RLC* teve os melhores resultados em 10 dos 12 casos experimentais (ver cor verde na tabela). No entanto, nos outros dois casos, a *RLC* foi muito competitiva com a *GNAT* (na *JAFFE*, com o 1º raio) e com a *LC* (na *AT&T*, com o 1º raio), onde a diferença de percentagens da média do nº de computações efectuadas em relação ao tamanho da base de dados é inferior a 1.3%

Em relação às estruturas de dados métricas dinâmicas (*RLC* e *HDSAT2*), verificamos que a *RLC* teve um desempenho superior em todas as bases de dados.

A *VPTree* foi estrutura de dados métrica que obteve o pior desempenho em todos os casos experimentais.

Em relação à degradação do desempenho das estruturas de dados, entre o primeiro e o terceiro raio de pesquisa, verificamos que as que se degradaram mais foram a *VPTree* e a *LAESA* nas bases de dados *Faces94*, *JAFFE* e *AT&T*, e a *GNAT* e *VPTree* na base de dados *Yalefaces*. A *RLC* foi a estrutura que apresentou a menor degradação nas diversas bases de dados, excepto na *Yalefaces* onde a *LAESA* foi a que se degradou menos.

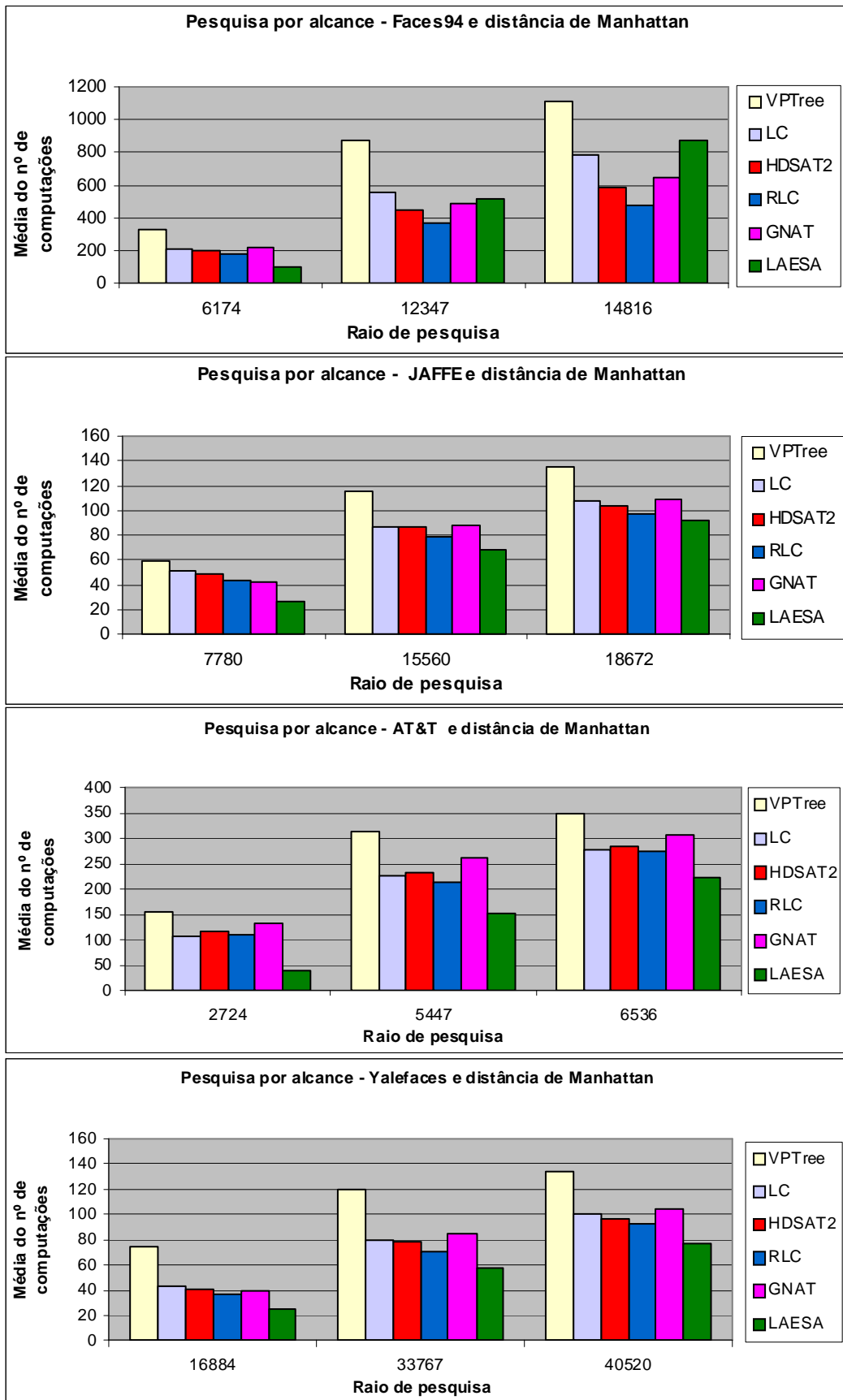


Figura 8.3 – Média do nº de computações das várias estruturas de dados métricas nas diferentes bases de dados e com a distância de *Manhattan*

### 8.4.3. Avaliação das estruturas de dados com a *Mahalanobis – LI*

O gráfico da figura 8.4 mostra, para as diferentes bases de dados e diversos raios de pesquisa, a média do número de computações de distâncias realizadas nas pesquisas por alcance com as diferentes estruturas de dados métricas, tendo como métrica a *Mahalanobis – LI*. A tabela 8.11 apresenta a percentagem da média do nº de computações efectuadas com as diversas estruturas de dados métricas em relação ao tamanho das várias bases de dados com a *Mahalanobis – LI*.

	<i>Faces94</i>			<i>JAFFE</i>			<i>AT&amp;T</i>			<i>Yalefaces</i>		
	1º raio	2º raio	3º raio	1º raio	2º raio	3º raio	1º raio	2º raio	3º raio	1º raio	2º raio	3º raio
<i>VPtree</i>	15,52%	39,64%	47,38%	41,36%	68,2%	77,7%	53,33%	88,64%	93,85%	64,71%	88,61%	94,1%
<i>LC</i>	7,54%	22,73%	31,35%	31,29%	53,38%	63,09%	33,91%	65,95%	78,51%	35,6%	68,3%	80,92%
<i>HDSAT2</i>	8,31%	18,14%	23,81%	30,31%	54,49%	64,5%	34,51%	70,46%	82,37%	36,38%	67,99%	79%
<i>RLC</i>	7,46%	14,02%	18,35%	28,32%	48,75%	61,88%	32,8%	65,11%	81,24%	32,03%	67,8%	81,65%
<i>GNAT</i>	10,31%	21,28%	27,18%	30,99%	55,58%	66,12%	45,46%	78,15%	87,15%	41,44%	76,88%	87,09%
<i>LAESA</i>	2,73%	23,82%	38,79%	19,03%	43,62%	53,13%	13,38%	50,69%	67,79%	24,57%	58,32%	72,66%

Tabela 8.11 – Percentagem da média do nº de computações efectuadas em relação ao tamanho da base de dados com a *Mahalanobis – LI*

Os resultados apresentados mostram que, face à pesquisa exaustiva, houve uma redução do número de computações da métrica entre 52,62% a 97,27% na base de dados *Faces94*, entre 22,3% a 80,97% na *JAFFE*, entre 6,15% a 86,62% na *AT&T* e entre 5,9% a 75,43% na *Yalefaces*.

Em relação às estruturas de dados métricas, a *LAESA* foi a que obteve os melhores resultados. Neste caso, dos 12 casos experimentais verificamos que a *LAESA* foi a melhor em 10 e apenas foi superada pela *RLC* na base de dados *Faces94* na pesquisa por alcance com o segundo e terceiro raio de pesquisa (ver cor amarela, na tabela 8.11).

Excluindo a *LAESA*, a *RLC* foi a estrutura de dados que obteve os segundos melhores resultados em 10 de 12 casos experimentais. Neste caso, verificamos que a *RLC* apenas não foi a melhor na base de dados *AT&T* com o terceiro raio de pesquisa onde foi superada pela *LC*, e na *Yalefaces* com o terceiro raio de pesquisa onde a *HDSAT2* foi superior.

Em relação às estruturas de dados métricas dinâmicas (*RLC* e *HDSAT2*), verificamos que a *RLC* teve um desempenho superior em todos os casos experimentais, excepto na base de dados *Yalefaces* com o terceiro raio de pesquisa.

A *VPtree* teve o pior desempenho em todos os casos experimentais.

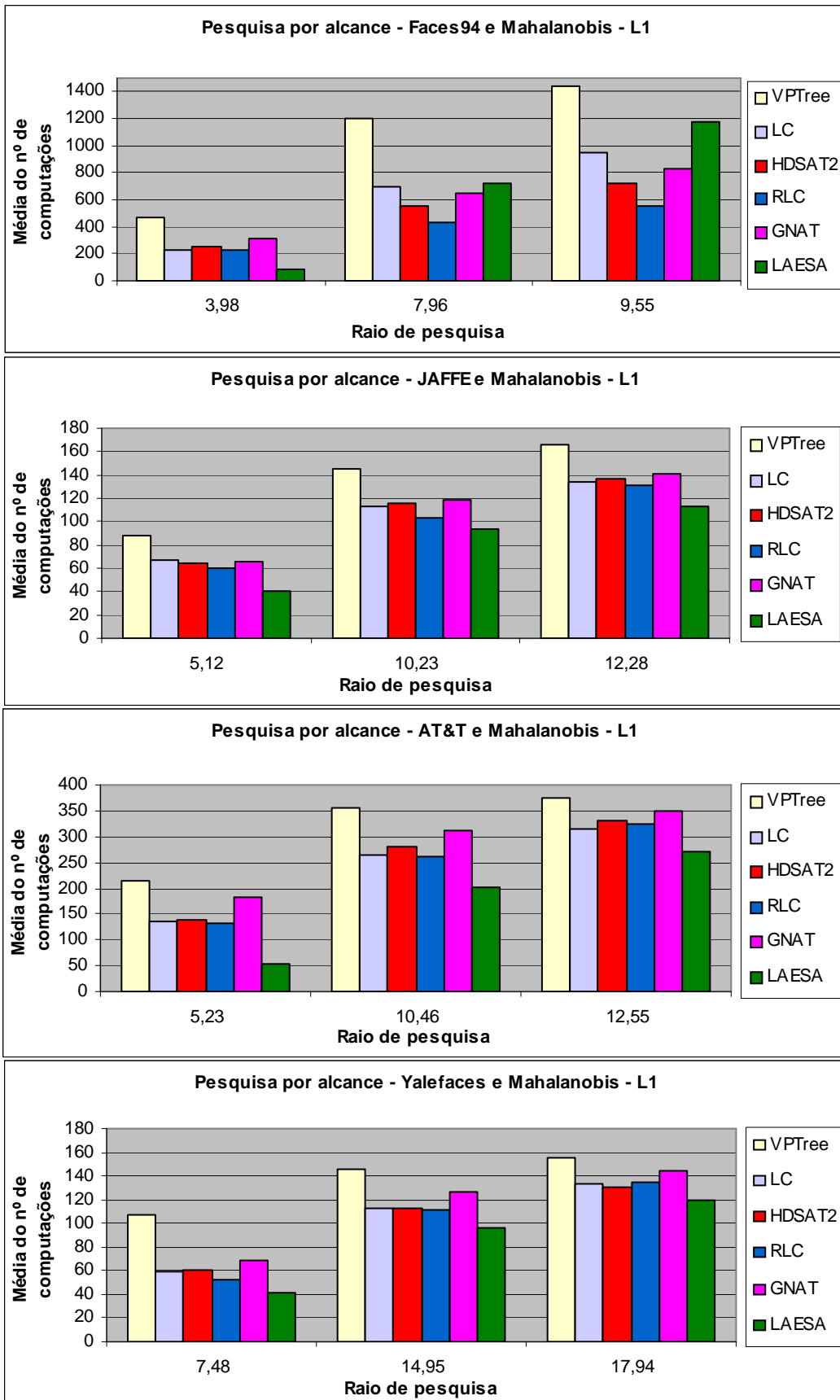


Figura 8.4 – Média do nº de computações das várias estruturas de dados métricas nas diferentes de dados e com a Mahalanobis – L1

Em relação à degradação do desempenho das estruturas de dados, entre o primeiro e o terceiro raio de pesquisa, verificamos que as que se degradaram mais foram a *LAESA* e a *VPtree* na base de dados *Faces94*, a *VPtree* e a *GNAT* na *JAFFE*, a *LAESA* e a *RLC* na *AT&T* e a *RLC* e *LAESA* na *Yalefaces*. A *RLC* e a *LC* foram as estruturas que apresentaram, respectivamente, a menor degradação nas bases de dados *Faces94* e *JAFFE* e a *VPtree* foi a que degradou menos nas bases de dados *AT&T* e *Yalefaces*.

#### 8.4.4. Avaliação das estruturas de dados com a *Mahalanobis – L2*

O gráfico da figura 8.5 mostra, para as diferentes bases de dados e diversos raios de pesquisa, a média do número de computações de distâncias realizadas nas pesquisas por alcance com as diferentes estruturas de dados métricas, tendo como métrica a *Mahalanobis – L2*. A tabela 8.12 apresenta a percentagem da média do nº de computações efectuadas com as diversas estruturas de dados métricas em relação ao tamanho das várias bases de dados.

	<i>Faces94</i>			<i>JAFFE</i>			<i>AT&amp;T</i>			<i>Yalefaces</i>		
	1º raio	2ºraio	3º raio	1º raio	2º raio	3º raio	1ºraio	2ºraio	3º raio	1ºraio	2ºraio	3ºraio
<i>VPtree</i>	15,37%	38,8%	47,06%	38,86%	70,6%	79,09%	52,4%	88,39%	94,03%	67,1%	91,14%	94,89%
<i>LC</i>	7,3%	22,1%	31%	33,08%	57,2%	65,77%	34,48%	67,96%	79,97%	39,95%	70,06%	84,46%
<i>HDSAT2</i>	8,28%	17,75%	23,34%	32,75%	56,53%	67,2%	35,61%	70,34%	82,19%	37,14%	69,94%	80,63%
<i>RLC</i>	7,58%	14,15%	18,64%	30,61%	51,08%	65,24%	33,41%	65,22%	81,61%	36,7%	68,19%	82,54%
<i>GNAT</i>	9,16%	19,57%	25,46%	33,32%	59,21%	70,08%	44,49%	78,46%	87,03%	46,96%	79,63%	89,57%
<i>LAESA</i>	3,85%	32,08%	45,22%	20,32%	43,62%	53,52%	18,34%	58,82%	73,65%	23,74%	62,17%	75,88%

Tabela 8.12 – Percentagem da média do nº de computações efectuadas em relação ao tamanho da base de dados com a *Mahalanobis – L2*

Estes resultados mostram que, face à pesquisa exaustiva, houve uma redução do número de computações da métrica entre 52,94% a 96,15% na base de dados *Faces94*, entre 20,91% a 79,68% na *JAFFE*, entre 5,97% a 81,66% na *AT&T* e entre 5,11% a 76,26% na *Yalefaces*.

Em relação às estruturas de dados métricas, a *LAESA* foi a que obteve os melhores resultados na maioria dos casos. Dos 12 casos experimentais verificamos que a *LAESA* foi a melhor em 10. Nos outros dois casos, correspondentes à base de dados *Faces94* com o segundo e terceiro raio de pesquisa, foi a *RLC* que obteve o melhor desempenho (ver cor amarela na tabela 8.12).

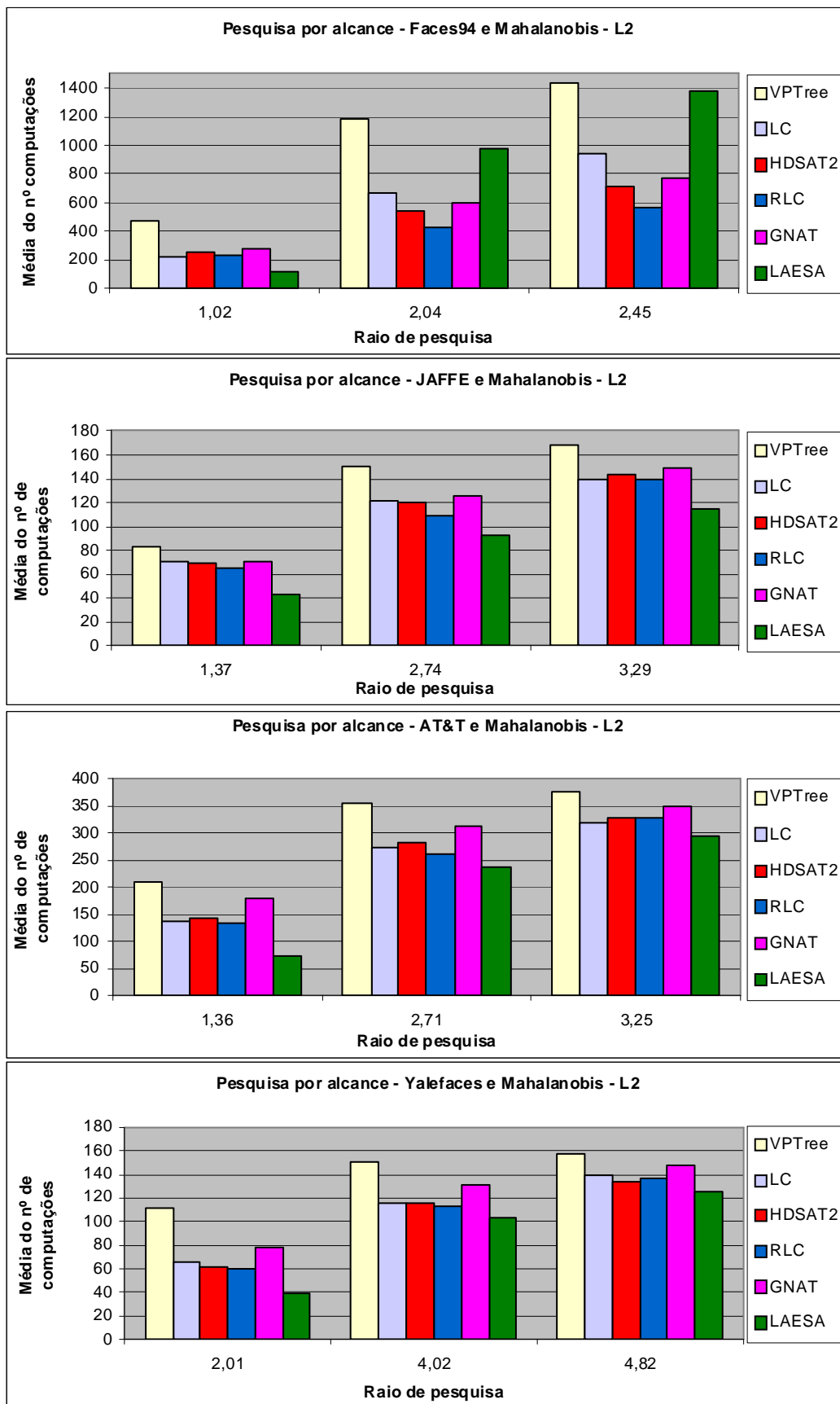


Figura 8.5 – Média do nº de computações das várias estruturas de dados métricas nas diferentes bases de dados e com a *Mahalanobis - L2*

Excluindo a *LAESA*, a *RLC* foi a estrutura de dados que teve os melhores resultados e só foi superada pela *LC* na base de dados *Faces94* com o primeiro raio de pesquisa e pela *HDSAT2* na base de dados *Yalefaces* com o terceiro raio de pesquisa.

Em relação às estruturas de dados dinâmicas verificamos que a *RLC* foi a melhor em 11 de 12 casos experimentais.

É de notar que a *VPtree* foi a pior de todas as estruturas de dados em todos os casos experimentais.

Em relação à degradação do desempenho das estruturas de dados, entre o primeiro e o terceiro raio de pesquisa, verificamos que as que se degradaram mais foram a *LAESA* e a *VPtree* na base de dados *Faces94*, a *VPtree* e a *GNAT* na *JAFFE*, a *LAESA* e a *RLC* na *AT&T* e a *LAESA* e *RLC* na *Yalefaces*. A *RLC*, *LC*, *GNAT* e *VPtree* foram as estruturas que apresentaram a menor degradação respectivamente nas bases de dados *Faces94*, *JAFFE*, *AT&T* e *Yalefaces*.

## 8.5. Conclusão

Neste capítulo foram apresentados os resultados da avaliação das diversas estruturas de dados métricas na pesquisa por alcance das imagens de rosto semelhantes.

De modo a realizar uma avaliação “justa”, foi seleccionado o raio de pesquisa mais “apropriado” para a aplicação em causa e foram parametrizadas as estruturas de dados métricas de forma a obter os melhores resultados. Com base nesta análise, na avaliação final foram realizadas pesquisas por alcance com 3 raios de pesquisa para cada base de dados sobre um espaço métrico, utilizando todas as estruturas de dados métricas.

Dada a relevância do dinamismo em aplicações de imagens de rosto semelhantes, é de salientar que em relação aos resultados da avaliação, verificamos que a *LAESA* foi a estrutura de dados que obteve o melhor desempenho com as diferentes métricas nas várias bases de dados. Em segundo lugar, aparece, na maioria dos casos, a estrutura de dados métrica *RLC*, no entanto apresentou uma degradação significativa nas bases de dados de pequena dimensão em alguns espaços métricos. É de notar que a *VPtree* foi a estrutura de dados que obteve o pior desempenho.

Em relação às estruturas de dados métricas dinâmicas (*RLC* e *HDSAT2*), a *RLC* foi a que obteve os melhores resultados na maioria dos casos.

Os resultados desta experiência mostram que todas as estruturas de dados métricas reduziram o número de computações da métrica em relação ao tamanho da base de dados e, desta forma, são uma mais valia na pesquisa por alcance das imagens de rosto

semelhantes.

É de notar que face à pesquisa exaustiva houve uma redução entre 52,62% e 97,27 na base de dados *Faces94*, entre 20,91% a 87,77% na *JAFFE*, entre 5,97 % a 90,44% na *AT&T* e entre 5,11% a 86,24% na *Yalefaces*. Neste caso, verificamos que a base de dados com o maior número de elementos (*Faces94*) foi onde se conseguiu obter a maior redução do número de computações da métrica.

Em relação aos espaços métricos que apresentaram a melhor performance na avaliação efectuada no capítulo anterior (Euclidiana e *Manhattan*) verificamos que, na maioria dos casos, a *LAESA* e *RLC* foram as estruturas de dados métricas que obtiveram o melhor desempenho nestes espaços métricos.

## Capítulo 9. Protótipo

Neste capítulo será apresentado o protótipo final, realizado durante a fase de elaboração da dissertação. O protótipo foi implementado em Java, e permite realizar pesquisas por alcance de imagens de rosto. Desta forma, dada uma imagem de rosto de consulta e um raio de pesquisa, o sistema devolve o conjunto de imagens de rosto semelhantes/próximas que existem na base de dados seleccionada, de acordo com a pesquisa realizada.

Este protótipo foi realizado com o objectivo de poder avaliar um conjunto de estruturas de dados métricas nas pesquisas por alcance em 4 bases de dados (*Faces94*, *JAFFE*, *AT&T* e *Yalefaces*). Para além disso, é possível escolher a medida de semelhança a utilizar em cada pesquisa. As medidas de semelhança implementadas foram: a distância Euclidiana, a distância de *Manhattan*, a *Mahalanobis-L1* e a *Mahalanobis-L2*.

Este capítulo está estruturado em 3 secções. Na secção 1 é apresentada a interface e os requisitos funcionais do protótipo. Na secção 2 é apresentada a arquitectura do sistema. A última secção diz respeito à conclusão.

### 9.1. Interface e funcionalidades

O ponto de entrada na aplicação apresenta o menu representado na figura 9.1.

Na interface do protótipo existem 4 componentes de visualização (CV1, CV2, CV3 e CV4), 5 menus de parametrização (MP1, MP2, MP3, MP4 e MP5) e dois botões (“Perform” e “Statistics”).

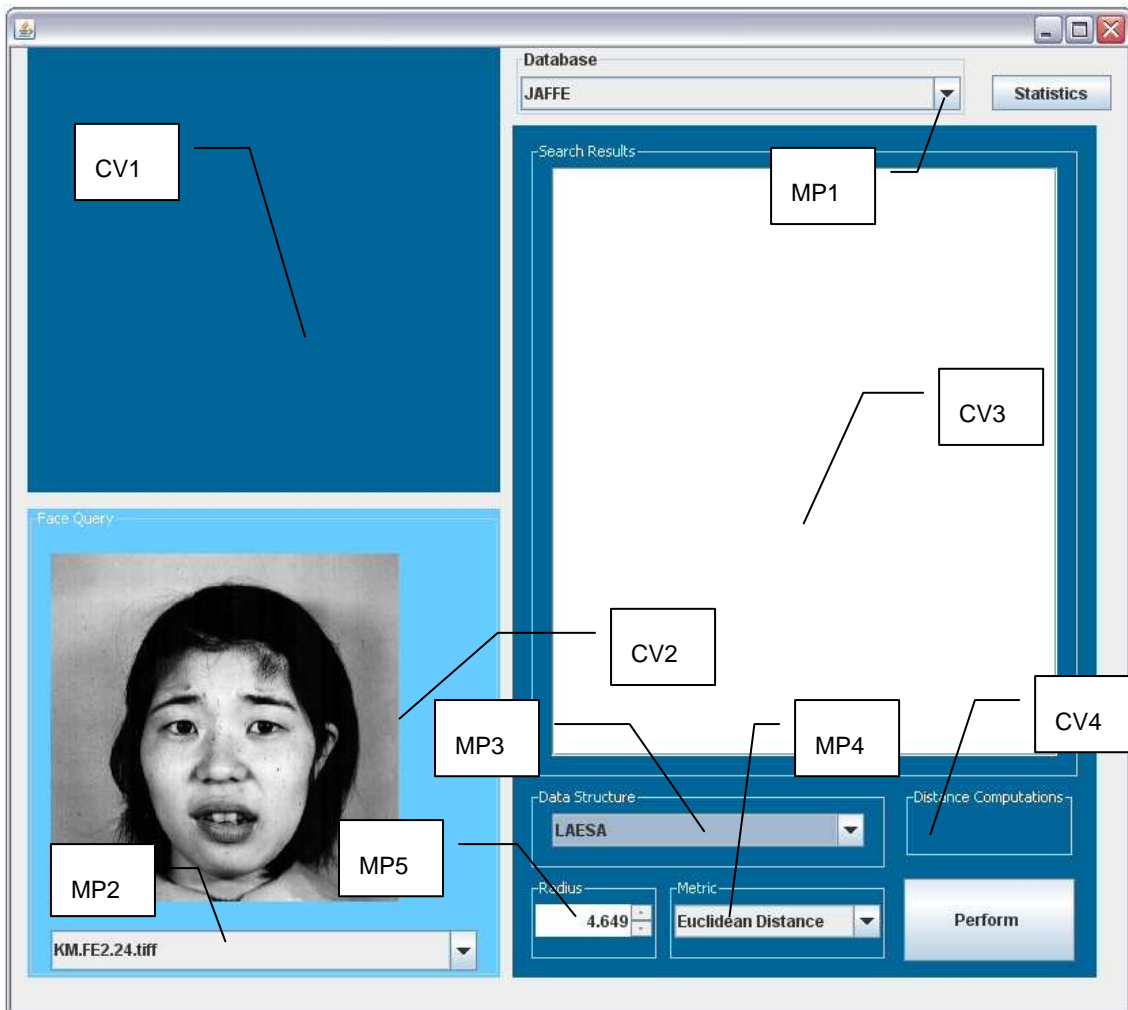


Figura 9.1 – Descrição dos componentes da interface

De seguida será feita uma descrição das principais funcionalidades e, por conseguinte, serão explicados os componentes de visualização e menus presentes na figura 9.1.

### 9.1.1. Pesquisa por alcance

Para realizar uma pesquisa por alcance, o utilizador deve: seleccionar a base de dados (MP1), a imagem de rosto de consulta (MP2), a estrutura de dados (MP3), a métrica (MP4) e o raio de pesquisa (MP5).

A selecção da base de dados é realizada no menu de parametrização MP1 que tem como opções as bases de dados *JAFFE*, *Faces94*, *AT&T* e *Yalefaces*. Sempre que uma base de dados é seleccionada no menu MP1, o menu MP2 é actualizado, tendo como elementos todas as imagens de rosto associadas à base de dados. Neste momento, o

componente de visualização CV2 também é actualizado com a primeira imagem de rosto que pertence à base de dados. Este componente servirá para indicar a imagem de rosto que será usada para realizar a pesquisa. A selecção da imagem de consulta é feita no menu MP2. Uma vez escolhida a imagem de rosto para consulta, o componente de visualização CV2 é actualizado com a imagem seleccionada.

Com os menus MP3 e MP4, o utilizador tem a possibilidade de seleccionar a estrutura de dados métrica e a métrica que será utilizada na pesquisa. As estruturas de dados métricas disponíveis no menu MP3 são: *LAESA*, *VPTree*, *GNAT*, *DSAT*, *HDSAT1*, *HDSAT2*, *LC* e *RLC*. O menu MP4, que permite seleccionar a métrica, tem como opções as distâncias: Euclidiana, *Manhattan*, *Mahalanobis - L1* e *Mahalanobis - L2*.

A selecção do raio que será utilizado na pesquisa por alcance é realizada no menu MP5. Por defeito, uma vez escolhida a métrica aparece o raio (em MP5) que obteve o melhor desempenho na pesquisa por semelhança (secção 8.1.2).

Para executar uma pesquisa por alcance, o utilizador deve pressionar o botão “Perform”. Com base nos parâmetros seleccionados, é realizada uma pesquisa por alcance das imagens de rosto semelhantes em relação à imagem de consulta seleccionada em MP2 e ao raio indicado em MP5. Esta pesquisa é realizada com a estrutura de dados métrica seleccionada em MP3. Uma vez executada essa pesquisa, no componente de visualização CV3, é apresentada uma listagem das imagens de rosto que fazem parte do conjunto de resultados. Para visualizar cada uma das imagens da listagem, o utilizador deve clicar no nome da imagem de modo a aparecer no componente de visualização CV1 a imagem requerida. No componente de visualização CV4 é apresentado o número de computações de distâncias efectuadas nessa pesquisa com a estrutura de dados métrica seleccionada.

A figura 9.2 apresenta a interface do programa depois de ter sido executado a pesquisa por alcance para a imagem de rosto com o nome “KM.FE2.24.jpg” da base de dados *JAFFE*. Neste exemplo, foi utilizada a estrutura de dados *VPTree*, a métrica *Mahalanobis - L1* e um raio de pesquisa de 12,24. Nesta pesquisa foi necessário executar 155 computações da medida de semelhança e teve como resultados 18 imagens.

Neste exemplo de pesquisa por alcance, pode-se verificar que todos os elementos da lista de resultados correspondem a imagens de rosto da mesma pessoa, que foi usada como consulta. Isto é facilmente visível já que as imagens da base de dados com o

mesmo prefixo estão associadas à mesma identidade.

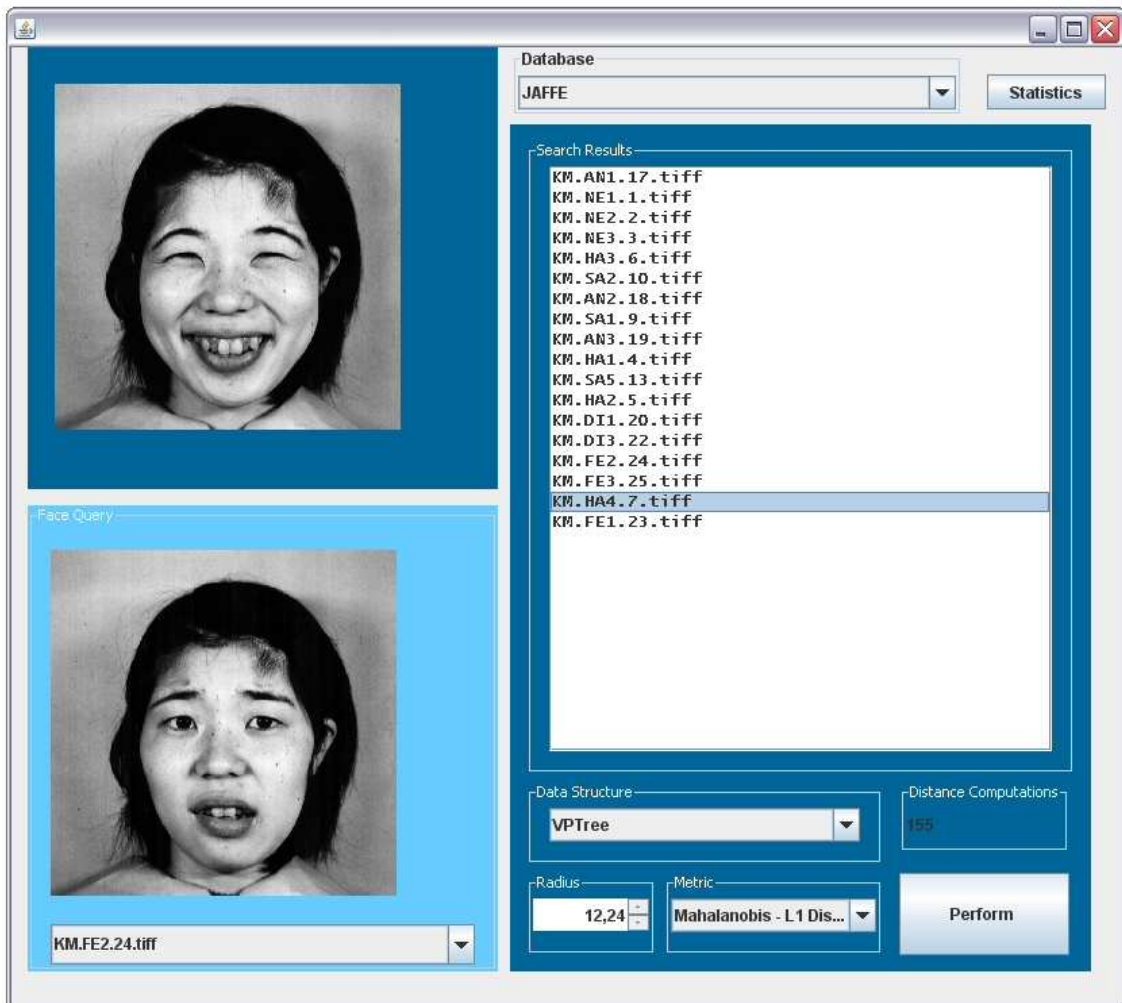


Figura 9.2 – Exemplo da execução de uma pesquisa

### 9.1.2. Estatísticas

O botão “Statistics” serve para obter as estatísticas relacionadas com as avaliações que foram efectuadas sobre as estruturas de dados métricas. Neste caso, ao clicar neste botão é exibida uma nova janela, com uma área de texto, onde são apresentadas as estatísticas referentes a:

- Raios de pesquisa: Calcula os raios de pesquisa mais pertinentes para cada espaço métrico e as respectivas médias das taxas de verdadeiros positivos, falsos positivos e do número de elementos retornados das várias pesquisas por alcance (ver secção 8.1.2).
- Parametrização das estruturas de dados métricas: Apresenta o

comportamento de cada estrutura de dados aquando da variação dos parâmetros de construção, nos diferentes espaços métricos (ver secção 8.2 e anexo A3). Para cada parametrização associada a uma determinada estrutura de dados é apresentado a média do número de computações da métrica das diversas pesquisas por alcance.

- Pesquisa por alcance: Apresenta os resultados da avaliação final das várias estruturas de dados métricas nos diferentes espaços métricos (ver secção 8.4).

Estas estatísticas apenas dizem respeito à base de dados que se encontra carregada no momento.

## 9.2. Arquitectura do sistema

Na figura 9.3 é apresentado o diagrama de pacotes do protótipo que foi implementado. Como podemos verificar pela figura, a aplicação está organizado em 2 pacotes designados “frEigenfaces” e “dataStructures”.

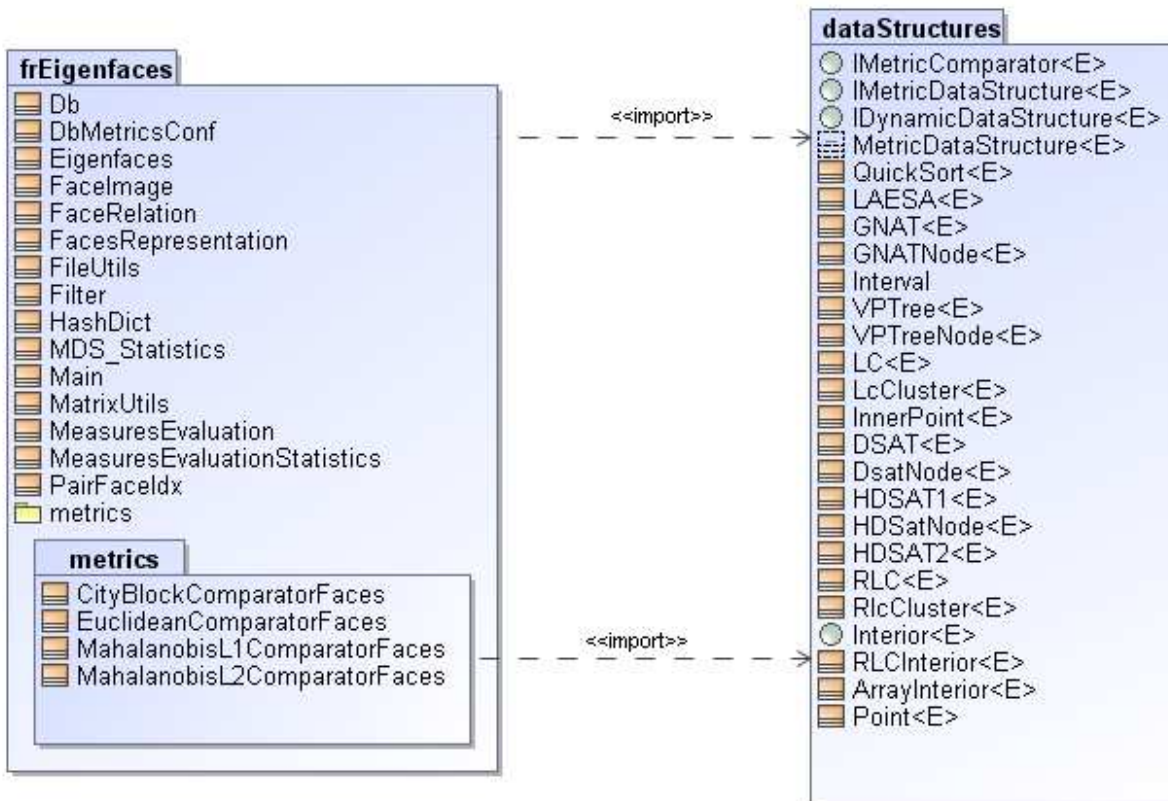


Figura 9.3 – Diagrama de pacotes do protótipo

O pacote “frEigenfaces” contém a implementação da interface principal da aplicação, do método de *Eigenfaces*, das várias métricas, e das várias estatísticas associadas às diferentes métricas e estruturas de dados. Na tabela 9.1 são apresentadas as classes consideradas mais relevantes que pertencem ao pacote “frEigenfaces”.

O sub-pacote “metrics” pertence ao pacote “frEigenfaces” e contém a implementação das várias métricas utilizadas para medir a similaridade entre duas imagens de rosto (ver secção 7.1.2).

O pacote designado de “dataStructures” contém a implementação das várias estruturas de dados métricas que foram usadas neste trabalho (ver capítulo 5): *LAESA*, *VPTree*, *GNAT*, *DSAT*, *HDSAT1*, *HDSAT2*, *LC* e *RLC*.

<b>Classe</b>	<b>Descrição</b>
Eigenfaces	Corresponde à implementação do método de <i>Eigenfaces</i> .
FaceImage	Corresponde à representação reduzida de uma imagem de rosto, depois de aplicado o método de <i>Eigenfaces</i> .
FacesRepresentation	É a classe responsável por carregar a base de dados e criar as respectivas representações das imagens de rosto.
DB	É a classe que representa uma base de dados.
MeasuresEvaluationStatistics	Corresponde à classe que foi utilizada para obter as várias estatísticas relacionadas com a avaliação das diferentes métricas.
MDS_Statistics	Corresponde à classe que foi utilizada para obter as várias estatísticas relacionadas com as estruturas de dados métricas.
Main	Corresponde à interface principal da aplicação.

Tabela 9.1 – Descrição das classes mais relevantes do pacote “frEigenfaces”

Neste caso, as estruturas de dados métricas foram parametrizadas com as mesmas configurações que foram escolhidas para a avaliação final das estruturas de dados métricas documentada no capítulo 8.

Nesta secção descreve-se a organização do protótipo no que diz respeito às bases de dados utilizadas e à implementação das suas funcionalidades, nomeadamente o carregamento da informação e o processamento da listagem.

### 9.2.1. Bases de dados utilizadas

O conjunto de bases de dados utilizadas no protótipo final é composto pela *JAFFE*, *Faces94*, *AT&T* e *Yalefaces* e encontra-se armazenado na directoria “/DBs”. Esta pasta encontra-se na mesma directoria em que o programa principal é executado.

Para cada base de dados foi criada uma directoria com o nome igual ao da base de dados. Esta directoria contém o conjunto de imagens de rosto associado à base de dados.

Pelo facto de ter sido utilizado o método de *Eigenfaces*, para a representação das imagens de rosto, é necessário ter para cada base de dados um conjunto de imagens de rosto de treino. Desta forma, cada directoria associada a uma base de dados contém uma outra com o nome “trainingSet” onde é armazenado o conjunto de imagens de rosto de treino.

### 9.2.2. Carregamento da informação

Cada base de dados contém um ficheiro no qual são guardadas as representações reduzidas de cada imagem de rosto. Este ficheiro tem o nome “faces.txt” e encontra-se na directoria associada à base de dados. Logo, na inicialização do protótipo é carregado o conjunto de representações reduzidas de cada imagem de rosto associado à base de dados seleccionada. Cada imagem de rosto é representada pela classe *FaceImage*.

No caso do ficheiro “faces.txt” não existir é necessário criar a representação das imagens de rosto, ou seja, aplicar o método de *Eigenfaces* para obter uma representação das várias imagens de rosto associadas à base de dados, o qual é realizado pelas classes *FacesRepresentation* e *Eigenfaces*.

O conjunto de instâncias da classe *FaceImage*, obtido do processo anterior, é armazenado num vector. Este conjunto é guardado em disco no ficheiro “faces.txt” na directoria da base de dados que foi carregada.

Com base no conjunto de instâncias da classe *FaceImage* é realizada a inicialização das estruturas de dados métricas. Como em algumas estruturas de dados a ordem de inserção determina a organização dos vários elementos na estrutura, é usada uma permutação da base de dados. Desta forma, para cada espaço métrico são construídas as várias estruturas de dados métricas com esta permutação das instâncias.

Este processo de carregamento da informação da base de dados é realizado sempre que é escolhida uma nova base de dados no menu MP1.

### **9.3. Conclusão**

Neste capítulo foi descrito o protótipo final que permite realizar pesquisas por alcance das imagens de rosto com as várias estruturas de dados métricas.

O protótipo apresentado foi utilizado para a tarefa de parametrização das diversas estruturas de dados métricas nas diferentes bases de dados sobre os espaços métricos usados. Também foi utilizado para gerar as várias estatísticas que permitiram avaliar os diferentes espaços métricos e analisar o comportamento das várias estruturas de dados nas pesquisas por alcance.

Em relação à interface gráfica, verificamos que as pesquisas nas várias estruturas de dados podem ser testadas em diferentes bases de dados e espaços métricos. Neste caso, para cada pesquisa o utilizador tem a possibilidade de especificar o raio de alcance pretendido.

Os resultados de cada pesquisa por alcance são apresentados em forma de lista com o nome dos ficheiros de imagens de rosto. Neste caso, é possível visualizar cada uma das imagens de rosto do conjunto de resultados. Além disso, é apresentado o número de computações da métrica que foram necessárias para realizar a pesquisa por alcance.

## Capítulo 10. Conclusão e trabalho futuro

Neste capítulo serão apresentadas as conclusões finais deste trabalho e serão indicadas direcções para trabalho futuro.

### 10.1. Conclusão

De modo a analisar a aplicabilidade das estruturas de dados métricas no domínio das imagens de rosto, neste trabalho foi realizado:

- Uma descrição dos diferentes métodos de representação de imagens de rosto e foram identificados os trabalhos onde foram usados;
- Uma descrição das várias medidas de semelhança usadas para comparar imagens de rosto e foram identificados os trabalhos onde foram usados;
- Implementação e avaliação de 4 espaços métricos em 4 bases de dados (*Faces94*, *JAFFE*, *AT&T* e *Yalefaces*). Os espaços métricos usados utilizam o método de *Eigenfaces* para a representação de imagens de rosto e as métricas: *Euclidiana*, *Manhattan*, *Mahalanobis – L1* e *Mahalanobis – L2*.

Desta análise pode-se concluir que é possível utilizar estruturas de dados métricas no domínio das imagens de rosto, obtendo resultados nas pesquisas que se consideram “satisfatórios” neste domínio de aplicação.

De modo a analisar a eficiência das pesquisas por alcance nas estruturas de dados métricas no domínio das imagens de rosto, neste trabalho foi realizado:

- Uma descrição das estruturas de dados métricas e implementação do pacote de estruturas de dados métricas. Este pacote é composto pela *LAESA*, *VPTree*, *GNAT*, *LC*, *DSAT*, *HDSAT1*, *HDSAT2* e *RLC*.
- Uma avaliação da pesquisa por alcance nas 8 estruturas de dados métricas implementadas em 4 bases de dados e sobre 4 espaços métricos.

Desta análise pode-se concluir que as estruturas de dados métricas são uma mais valia na pesquisa por alcance das imagens de rosto semelhantes. Em todos os casos experimentais existiu uma redução do número de computações da métrica face à pesquisa exaustiva. A estrutura de dados métrica *LAESA* foi a que obteve o melhor desempenho seguida da *RLC*. Em relação às estruturas de dados métricas dinâmicas a *RLC* foi a que obteve os melhores resultados.

É de salientar que todas as avaliações deste trabalho foram realizadas com 4 bases de dados de imagens de rosto (*Faces94*, *JAFFE*, *AT&T* e *Yalefaces*) que já foram utilizadas em outros trabalhos relacionados com o reconhecimento de imagens de rosto. Estas bases de dados têm diferentes características em relação às condições de iluminação, variações de expressão e rotação, entre outros aspectos.

Para fazer as avaliações, foi desenvolvido um protótipo em JAVA que permite realizar pesquisas por alcance de imagens de rosto com as várias estruturas de dados métricas, nas bases de dados mencionadas.

No que diz respeito aos contributos deste trabalho, existem vários aspectos que gostaria de destacar:

- Face à pesquisa bibliográfica realizada, podemos concluir que este é o primeiro trabalho que usa estruturas de dados métricas para a pesquisa por alcance em imagens de rosto;
- A maioria dos trabalhos na área de reconhecimento de imagens de rosto não aborda os temas relacionados com a eficiência na pesquisa. Este trabalho mostra que é possível agilizar as pesquisas por alcance das imagens de rosto semelhantes;
- Foi realizada uma avaliação da eficiência da pesquisa por alcance com estruturas de dados métricas no domínio das imagens de rosto, no qual se comparam 8 estruturas de dados métricas, usando 4 métricas em 4 bases de dados;
- Foi realizada uma avaliação para as 4 métricas (distância Euclidiana, distância de *Manhattan*, *Mahalanobis - L1*, *Mahalanobis - L2*), utilizando o método de *Eigenfaces* para a representação das imagens de rosto, porque não foram encontrados trabalhos onde estas 4 métricas tivessem sido avaliadas.
- O pacote de estruturas de dados desenvolvido é genérico e, desta forma, poderá ser utilizado para estudar a aplicabilidade e eficiência das estruturas

de dados métricas noutros domínios de aplicação;

- Foi implementado um protótipo que permite realizar pesquisas por alcance das imagens de rosto semelhantes com as várias estruturas de dados métricas implementadas e que poderá ser a base para novos trabalhos nesta área.

Como conclusão final realça-se que é possível usar de forma eficiente estruturas de dados métricas nas pesquisas por alcance de imagens de rosto. No entanto, o desempenho destas técnicas só poderá ser realmente avaliado em aplicações concretas.

É também de salientar que este trabalho já deu origem a duas publicações [19] [43]. A primeira foi realizada na fase inicial desta dissertação e a segunda foi realizada na parte final.

Considera-se que o trabalho desenvolvido nesta área de imagens de rosto está longe de estar terminado pelo que, em seguida, se descrevem alguns tópicos que irão, provavelmente, ser considerados como temas de investigação futura.

## 10.2. Trabalho futuro

O trabalho que se propõe, vir a ser desenvolvido no futuro, está inserido em 2 tópicos:

- Técnicas eficientes para a pesquisa por semelhança;
- Representação/semelhança de imagens de rosto.

No que diz respeito às estruturas de dados métricas aqui implementadas e avaliadas, é fundamental:

1. Implementar a pesquisa dos  $k$  - vizinhos mais próximos;
2. Avaliar esta pesquisa no que diz respeito à sua eficiência em diferentes domínios;
3. Avaliar as pesquisas por semelhança considerando actualizações à base de dados (inserções e remoções).
4. Avaliar as pesquisas por semelhança considerando também a complexidade espacial.

No que diz respeito ao domínio de aplicação (imagens de rosto) é fundamental:

1. Identificar outros espaços métricos que utilizem outras formas de representação de imagens de rosto, como por exemplo com os descritores extraídos com a técnica *SIFT*;

2. Avaliar esses espaços métricos face aos implementados nesta tese;
3. Avaliar a eficiência da pesquisa por alcance com as estruturas de dados métricas nestes novos espaços métricos.

Com respeito ao protótipo desenvolvido é fundamental que este permita a inserção/remoção de imagens de rosto.

No que diz respeito à avaliação realizada nesta tese seria conveniente comparar outras técnicas que não sejam baseadas em espaços métricos com a avaliação aqui apresentada.

Por último, seria muito interessante poder usar as estruturas de dados métricas numa aplicação concreta na área das imagens de rosto.

## **Bibliografia**

- [1] - W. Zhao, R. Chellappa, P. Phillips, A. Rosenfeld, "Face Recognition: A literature Survey", ACM Computing Surveys, Vol. 35 (4), p.399 - 458, 2003
- [2] - M. Turk, A. Pentland, "Face Recognition using eigenfaces", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p. 586-591, 1991
- [3] - X. Tan, S. Chen, Z. Zhou, F. Zhang, "Face Recognition from a Single Image per Person: A Survey", Pattern Recognition, Vol. 39(9), p.1725-1745, 2006
- [4] - R. Beveridge, D. Bolme, M. Teixeira, B. Draper, "The CSU Face Identification Evaluation System User's Guide Version 5.0", Available: <http://www.cs.colostate.edu/evalfacerec/algorithms/version5/faceIdUsersGuide.pdf>
- [5] - W. Yambor, B. Draper, J. Beveridge, "Analyzing PCA-Based Face Recognition Algorithms: Eigenvector Selection and Distance Measures", Proc. Second Workshop Empirical Evaluation Methods in Computer Vision, 2000
- [6] - M. Mamede, "A dynamic data structure for range queries in high dimensional metric spaces". Submitted for publication, 2007. Available: [http:// ctp.di.fct.unl.pt/~mm /dynamic-07.pdf](http://ctp.di.fct.unl.pt/~mm/dynamic-07.pdf)
- [7] - M. Mamede, F. Barbosa, "Range Queries in Natural Language Dictionaries With Recursive Lists of Clusters", Proc. of 22th International Symposium on Computer and Information Sciences, 2007
- [8] - E. Vivek, N. Sudha, "Gray Hausdorff distance measure for comparing face

images”, IEEE Transactions on Information Forensics and Security, Vol. 1(3), p. 342-349, 2006

[9] - K. Delac, M. Grgic, T. Kos, “Sub-Image, Homomorphic Filtering Techniques for Improving Facial Identification under Difficult Illumination Conditions”, Proceedings of the International Conference on System Signals and Image Processing, 2006

[10] - M. Mendonça, J. Denipote, R.Fernandes, M. Paiva, “Illumination Normalization Methods for Face Recognition”, Proceedings of 20th Brazilian Symposium on Computer Graphics and Image Processing, 2007

[11] - H. Liu, W. Gao, J. Miao, J. Li, “A novel method to compensate variety of illumination in face detection”, Proceedings of the 6th Joint Conference on Information Sciences, p. 692 – 695, 2002

[12] - S. King, G. Tian, D. Taylor, S. Ward, “Cross-Channel Histogram Equalisation for Colour Face Recognition”, Lecture Notes in Computer Science, Vol. 2688, p. 1055-1056, 2003

[13] - S. Du, R. Ward, ”Wavelet-Based Illumination Normalization For Face Recognition”, IEEE International Conference on Image Processing, Vol. 2, p. 954-957, 2005

[14] - G. Huang, V. Jain, E. Learned-Miller, “Unsupervised Joint Alignmen of Complex Images”, Proceedings of the IEEE 11th International Conference on Computer Vision, p.1-8, 2007

[15] - Y. Russer, C. Tomasi, L. Guibas, “The Earth Mover’s Distance as a Metric for Image Retrieval”, International Journal of Computer Vision, Vol.40, p.99-121,1998

[16] - <https://face-recognizer.dev.java.net/>

[17] - <http://cswww.essex.ac.uk/mv/allfaces/index.html>

- [18] - D. Huttenlocher, G. Klanderman, W. Rucklidge, "Comparing Images Using the Hausdorff Distance", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15(9), p. 850-863, 1993
- [19] - F. Barbosa, F. Costa, P. Chambel, P. Paiva, "Pesquisa aproximada em diferentes domínios de aplicação", Proceedings of Conferencia Ibero-Americana WWW/Internet (CIAWI 2008), 2008
- [20] - A. Martínez, A. Kak, "PCA versus LDA", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23(2), p. 228 - 233, 2001
- [21] - P. Belhumeur, J. Hespanha, D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19(7), p. 711-720, 1997
- [22] - M. Bicego, A. Lagorio, E. Grosso, M. Tistarelli, "On the use of SIFT Features for Face Authentication", Conference on Computer Vision and Pattern Recognition Workshop, p. 35 - 35, 2006
- [23] - E. Chavez, G. Navarro, "A compact space decomposition for effective metric indexing", Pattern Recognition Letters, Vol. 26(9), p. 1363-1376, 2005
- [24] - P. Yianilos, "Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces", Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms, p. 311-321, 1993
- [25] - E. Chavez, G. Navarro, R. Baeza-Yates, J. Marroquín, "Searching in Metric Spaces", ACM Computing Surveys (CSUR), Vol. 33(3), p. 273-321, 2001
- [26] - M. Micó, J. Oncina, "A new version of the Nearest-Neighbour Approximation and Eliminating Search Algorithm (AESAs) with linear preprocessing time and memory requirements", Pattern Recognition Letters, Vol.15(1), p. 9-17, 1994
- [27] - S. Brin, "Near Neighbor Search in Large Metric Spaces", Proceedings of the 21st

International Conference on Very Large Data Bases, p. 574-584, 1995

[28] - D. Arroyuelo, F. Muñoz, G. Navarro, N. Reyes, “Memory-adaptative dynamic spatial approximation trees”, Proceedings of the 10th International Symposium on String Processing and Information Retrieval (SPIRE 2003), p.360- 368, 2003

[29] - T. Korenius, J. Laurikkala, M. Juhola, ”On principal Component Analysis, cosine and Euclidean measures in information retrieval”, Information Sciences, Vol. 177(22), p. 4893 - 4905, 2007

[30] - S. Sharma, “Applied multivariate techniques”, John Wiley & Sons, 1996

[31] - F. Barbosa, “Similarity-based retrieval in high dimensional data with Recursive Lists of Clusters: A study case with Natural Language Dictionaries”, to appear on Proc of International Conference on Information Management and Engineering (ICIME 2009)

[32] – H. Alam, F. Rahman, Y. Tarnikova, “A Pair-wise Decision Fusion Framework: Recognition of Human Faces”, Proceedings of the International Conference on Information Fusion (ISIF 2003). IEEE Xplore (ISBN:0-9721844-4-9), p. 1484-1489, 2003

[33] - <http://www.kasrl.org/jaffe.html>

[34] - M. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, “Coding facial expressions with Gabor wavelets”, Proceedings of the Third IEEE International Conference On Automatic Face and Gesture Recognition, p. 200-205, 1998

[35] - <http://www.cl.cam.ac.uk/research/dtg/attarchive/>

[36] - <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>

[37] - M. Aly, “Face Recognition using SIFT features”, Available:

[www.vision.caltech.edu/malaa/research/aly06face.pdf](http://www.vision.caltech.edu/malaa/research/aly06face.pdf), 2006

[38] - G. Navarro, N. Reyes, “Dynamic Spatial Approximation trees”, *Journal of Experimental Algorithmics (JEA)*, Vol. 12, 2008

[39] - G. Navarro, N. Reyes, “Fully Dynamic Spatial Approximation trees”, *Proceedings of the 9th International Symposium on String Processing and Information Retrieval*, p. 254-270, 2002

[40] - F. Barbosa, “Similarity-based retrieval in high dimensional data with Recursive Lists of Clusters: a study case with Natural Language Dictionaries”, *Proceedings of the International Conference on Information Management and Engineering (ICIME 2009)*, IEEE Computer Society (ISBN: 978-1-4244-3774-0), 2009

[41] - M. Mamede, “Recursive Lists of Clusters: A Dynamic Data Structure for Range Queries in Metric Spaces”, *Proceedings of the 20<sup>th</sup> International Symposium on Computer and Information Sciences (ISCIS 2005)*. Springer-Verlag, Berlin, Germany, pp. 843-853, 2005

[42] - F. Barbosa, A. Rodrigues, “Range Queries over Trajectory Data with Recursive List of Clusters: a case study with Hurricanes data”, *Geographic Information Science Research UK (GISRUK 2009)*, 2009

[43] - P. Chambel, F. Barbosa, “Improving Similarity Search in Face-Images Data”, Submitted to *Second Workshop on Very Large Digital Libraries (VLDL) of 13th European Conference on Research and Advanced Technologies on Digital Libraries (ECDL)*, 2009

[44] - <http://www-ctp.di.fct.unl.pt/~fb/pirs>



## **Anexos**

### **A 1. Normalização da informação**

Existem vários factores que degradam o desempenho dos sistemas no reconhecimento de imagens de rosto tais como variações das condições de iluminação, diferentes escalas, pose entre outras. De modo a contornar estes problemas podem ser aplicadas várias transformações nas imagens de rosto com o objectivo de minimizar estes efeitos. Este processo é denominado de normalização. Neste caso, existem dois tipos designados de normalização geométrica e fotométrica.

Na normalização geométrica são tratados os aspectos relacionados com a pose e variações de escala das imagens. O objectivo consiste em transformar os rostos de modo a que, no final do processo, estes se encontrem na mesma escala e posição. Estas transformações são importantes na medida em que permitem uniformizar as imagens no que diz respeito à forma, possibilitando, desta forma, um aumento na taxa de reconhecimento do sistema.

Em relação à normalização fotométrica são tratadas as condições relacionadas com a variação da iluminação. Desta forma, procura-se melhorar as condições da imagem de modo a tornar as comparações entre imagens menos sensíveis a diferentes condições de iluminação.

#### **A 1.1. Métodos baseados na normalização geométrica**

Nesta secção é feita uma breve descrição de um método de normalização geométrica designado de *Funneling*,

##### **A 1.1.1. *Funneling***

O alinhamento de objectos é um aspecto importante em sistemas de reconhecimento na medida em que o correcto posicionamento destes possibilita um aumento na taxa de reconhecimento. No trabalho referido em [14] é descrito um algoritmo de alinhamento para várias classes de objectos denominado de *funneling*. Desta forma, este algoritmo pode ser aplicado para corrigir o posicionamento de imagens de rosto.

Neste caso, com base num conjunto de treino de imagens de rostos é construído um mecanismo automático de alinhamento, sem que para tal seja necessário etiquetar as várias partes dos objectos ou identificar as poses iniciais destes. Assim, tendo como entrada uma imagem de rosto de exemplo, o sistema retorna uma versão alinhada desta.

## A 1.2. Métodos baseados na normalização fotométrica

Nesta secção é feita uma breve descrição de alguns métodos de normalização fotométrica tais como *Homomorphic Filter*, *LogAbout*, *Equalization Histogram* e um método baseado em *Wavelets*.

### A 1.2.1. Homomorphic Filter

O *Homomorphic Filter* é apresentado em [9] e [10] como um método que permite compensar as mudanças de iluminação de uma imagem de rosto. Este método é baseado na ideia de que a intensidade da luz reflectida pelos objectos é caracterizada por frequências espaciais altas e a intensidade da fonte de luz incidente no cenário por frequências espaciais baixas. A figura A 1.1 retirada de [9] descreve o funcionamento do *Homomorphic Filter*.

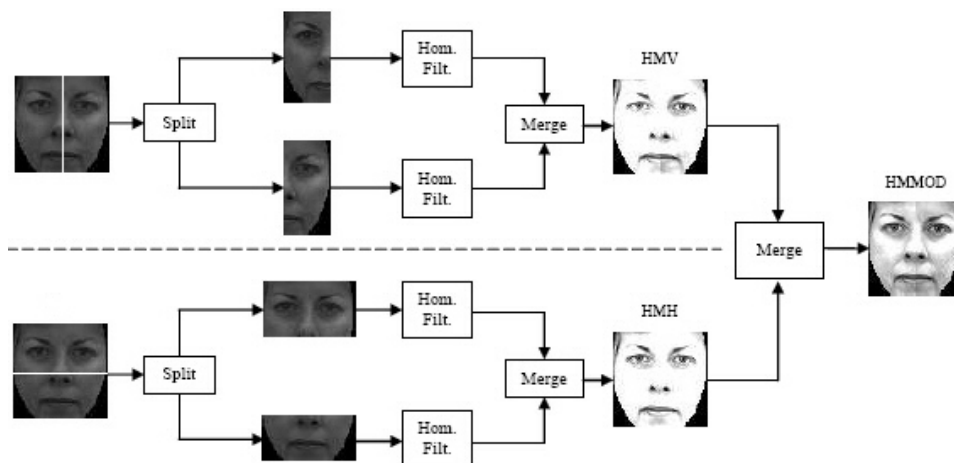


Figura A 1.1 – Esquema do funcionamento do *Homomorphic Filter*

Verificamos pelo esquema que é feita uma divisão vertical sobre a imagem original. De seguida é aplicado o filtro a cada metade e as sub-imagens resultantes são combinadas formando uma imagem completa. O filtro utilizado neste processo foi o

*Butterworth* passa-alto (termo em inglês, *high pass*) e realça as frequências altas e atenuando as baixas.

A mesma sequência é realizada sobre duas sub-imagens obtidas da imagem original através da divisão desta na horizontal. As duas imagens obtidas pelos dois processos anteriores são combinadas, tal como podemos verificar pela figura A 1.1 através da seguinte fórmula:

$$I_{HMMOD}(x, y) = \frac{1}{2} [I_{HNV}(x, y) + 0.75I_{HMH}(x, y)]$$

Neste caso,  $I_{HMMOD}$  corresponde à imagem final resultante da aplicação do filtro e  $I_{HNV}(x, y)$  e  $I_{HMH}(x, y)$  representam, respectivamente, a imagem dividida verticalmente e horizontalmente. Estas duas resultam do processo descrito anteriormente.

### A 1.2.2. *LogAbout*

Outro método usado para compensar a variação da iluminação é o *LogAbout* e foi apresentado em [11]. Este método consiste na aplicação de um filtro passa-alto seguido de uma transformação logarítmica. A fórmula que foi usada na transformação logarítmica é dada por:

$$g(x, y) = a + \frac{\ln(f(x, y) + 1)}{b \ln c}$$

Os parâmetros  $a$ ,  $b$  e  $c$  da fórmula definem a forma da curva logarítmica e  $f(x, y)$  corresponde à imagem.

A utilização do método sem a aplicação do filtro passa-alto sofria de uma possível perda parcial da informação das extremidades. É por essa razão que no *LogAbout* é aplicado o filtro passa-alto antes da transformação logarítmica com o objectivo de colmatar este problema.



Figura A 1.2 – Normalização com o método *LogAbout*

Na figura A1.2 (retirada de [11]) podemos verificar, na segunda imagem, a aplicação da transformação logarítmica à imagem original, e na última a aplicação do método *LogAbout*.

### A 1.2.3. *Histogram Equalization*

*Histogram Equalization* é um método bastante usado na área de reconhecimento e detecção de rostos como forma de aumentar o contraste da imagem na escala de cinzento. Um histograma consiste numa distribuição estatística dos níveis de cinzento de uma imagem e é usada para determinar a transformação em escala de cinzento que permite melhorar o contraste da imagem. O objectivo do método *Histogram Equalization* consiste em gerar uma imagem com níveis de cinzentos uniformes com base no histograma. A figura A1.3 (retirada de [11]) apresenta a aplicação do método *Histogram Equalization*.



Figura A 1.3 – Aplicação do *Histogram Equalization* sobre a primeira imagem

Podemos encontrar a referência ao método *Histogram Equalization* em [12], onde também é apresentada uma extensão deste aplicado a imagens com cores.

### A 1.2.4. *Wavelets*

Em [13] é apresentado um método de normalização baseado na transformação de *wavelets* que, além de melhorar o contraste, realça as extremidades das imagens de rostos. Este último aspecto não é tratado no método *Histogram Equalization* que apenas melhora o contraste.

Neste caso, é feita uma decomposição da imagem em componentes de alta e baixa frequência, sendo estes manipulados separadamente. Desta forma, o melhoramento do contraste é feito aplicando o método de *Histogram Equalization* aos coeficientes de

aproximação de baixa frequência. Em relação aos coeficientes de detalhe, caracterizados pelas altas-frequências, estes são multiplicados por um escalar ( $> 1$ ) de forma a realçar os detalhes da imagem. A imagem normalizada é obtida pela transformação *wavelet* inversa sobre os coeficientes modificados anteriormente.

Em [10] é feita uma comparação deste método com *LogAbout* e *Homomorphic Filter* e verificamos que o *Wavelet* foi o método que obteve o melhor desempenho em termos de taxa de reconhecimento. No entanto, o *LogAbout* também teve um desempenho próximo do *Wavelet*.



## A 2. Medidas de distância aplicadas a imagens

### A 2.1. EMD – *Earth Mover's Distance*

Em [15] é apresentada a *Earth Mover's Distance* como uma medida de semelhança que pode ser usada para a recuperação de imagens. Neste caso, para descrever e sintetizar as características das imagens são usadas distribuições de múltipla dimensão. Existem várias vantagens para transformar uma distribuição original por outra mais compacta, tais como, a redução do espaço necessário para armazenamento da informação, redução do tempo de processamento e a certeza de uma maior robustez perceptual na correspondência entre imagens. Assim, aparece o conceito de *signature* que representa uma distribuição de múltipla dimensão comprimida e de tamanho variável. Neste caso uma distribuição representada por um histograma é transformada numa “*signature*” que contém o conjunto de *clusters* dominantes da distribuição.

Neste caso, cada *histograma* é composto por um conjunto de *bins* em que cada *bin* representa um intervalo de valores que indica o número de vezes que um determinado intervalo de *pixels* aparece numa imagem. Assim, um histograma é convertido numa *signature* formado por vários *clusters*, onde cada *cluster* é formado por um elemento representativo e pelo peso que determina a fracção de *pixels* que pertence ao *cluster*. O elemento representativo consiste no valor central do intervalo que o *bin* representa.

Dadas duas *signatures* representativas de duas imagens, por analogia a primeira pode ser vista como um volume de terra espalhado num espaço e a segunda como um conjunto de buracos no mesmo espaço. Assim, a *Earth Mover's Distance* pode ser usada para medir o trabalho necessário para tapar esses buracos com a terra ou seja permite calcular o custo mínimo para transformar uma *signature* noutra.

Outro conceito importante está relacionado com a noção de medida entre as características básicas que estão agregadas dentro das distribuições ao qual damos o nome de *Ground distance*. Se esta for uma métrica e se a soma de todos os pesos de cada *signatures* for igual então a *Earth Mover's Distance* também é métrica.

A computação da *Earth Mover's Distance* pode ser tratada como um problema de transporte em que vários fornecedores abastecem vários consumidores. Desta forma, o problema de transporte consiste em encontrar os fluxos de bens, entre os fornecedores e os consumidores, de modo a minimizar o custo e a satisfazer as necessidades destes.

Para este problema o custo de transportar os bens entre cada entidade é conhecido.

Desta forma, a comparação entre *signatures* pode ser vista como o problema de transporte descrito anteriormente em que a primeira *signature* corresponde aos fornecedores e a segunda aos consumidores e o custo do transporte entre cada entidade é a *ground distance* entre os elementos de cada *signature*.

Sejam  $P = \{(p_1, w_{p_1}), \dots, (p_m, w_{p_m})\}$  e  $Q = \{(q_1, w_{q_1}), \dots, (q_n, w_{q_n})\}$  duas *signatures* com  $m$  e  $n$  *clusters* respectivamente onde  $p_i$  corresponde ao elemento representativo do *cluster* na posição  $i$  da *signature*  $P$ ,  $w_{p_i}$  é o peso associado ao *cluster*  $p_i$ ,  $D = [d_{ij}]$  a *ground distance* entre os *clusters*  $P_i$  e  $Q_j$ .

A *Earth Mover's Distance* pode ser formalizada como um problema de programação linear em que se pretende encontrar a matriz de fluxos  $F = [f_{ij}]$  em que  $f_{ij}$  representa o fluxo entre  $p_i$  e  $q_j$  que minimiza o custo global:

$$Work(P, Q, F) = \sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij}$$

Neste caso os fluxos estão sujeitos às seguintes restrições:

1.  $f_{ij} \geq 0 \quad 1 \leq i \leq m, 1 \leq j \leq n$
2.  $\sum_{j=1}^n f_{ij} \leq w_{p_i} \quad 1 \leq i \leq m$
3.  $\sum_{i=1}^m f_{ij} \leq w_{q_j} \quad 1 \leq j \leq n$
4.  $\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min(\sum_{i=1}^m w_{p_i}, \sum_{j=1}^n w_{q_j})$

A primeira restrição indica que os bens devem ser transferidos apenas de  $P$  para  $Q$ , ou seja, dos fornecedores para os consumidores. A segunda limita a quantidade de bens que pode ser enviada do *cluster* em  $P$  para  $Q$ , ou seja, a quantidade que cada fornecedor pode enviar. Este limite é dado pelo peso correspondente do *cluster* em  $P$ .

Na terceira restrição os *clusters* em  $Q$ , ou seja, os consumidores não podem receber mais bens do que aqueles que estes necessitam, ou seja, indicam a quantidade que cada consumidor pode receber. Este limite é dado pelo peso correspondente do *cluster* em  $Q$ . A última restrição obriga o transporte do máximo de bens possível.

Com a resolução do problema de transporte obtemos a matriz de fluxos óptima e,

desta forma, a *Earth Mover's Distance* é definida da seguinte forma:

$$EMD(P, Q) = \frac{\sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij}}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}}$$



### **A 3. Parametrização das estruturas de dados métricas**

A parametrização das diversas estruturas de dados métricas depende da base de dados sobre o espaço métrico. Como pretendíamos usar 4 bases de dados sobre os 4 espaços métricos foram realizados 16 casos experimentais por estrutura de dados.

Cada caso experimental envolve 3 pesquisas por alcance nas 3 permutações criadas da base de dados (ver secção 8.1.1), numa dada estrutura de dados métrica.

Para analisar o comportamento de uma estrutura de dados métrica associada a um determinado espaço métrico numa determinada base de dados foram testadas várias parametrizações. Para cada parametrização foram construídas três instâncias da estrutura de dados métrica, tendo como base as 3 permutações da base de dados e foram realizadas várias pesquisas por alcance nestas estruturas, que correspondem aos elementos pertencentes ao conjunto de consulta criado. Para as várias pesquisas por alcance foram utilizados os raios de pesquisa definidos em 8.1.2. Desta forma, para cada parametrização foi contabilizada a média do número de computações da métrica que foram efectuadas para realizar as diversas pesquisas, ou seja, para cada permutação foi calculada a média de distâncias calculadas das várias pesquisas realizadas com o conjunto de elementos de consulta e depois foi calculada a média destes valores nas 3 permutações.

Nesta secção serão apresentados alguns gráficos que foram usados no estudo dos parâmetros de construção das diversas estruturas de dados métricas. Como foi descrito no documento, as parametrizações usadas na avaliação final das estruturas de dados métricas e no protótipo foram escolhidas com base na análise destes gráficos.

#### **A 3.1. Parametrização da *LAESA***

Os gráficos da figura A3.1 apresentam o comportamento da *LAESA* aquando da variação do número de *pivots* nas diferentes bases de dados com as diferentes métricas.

Estes gráficos mostram uma descida significativa do número de computações da métrica com o aumento inicial do número de *pivots*. No entanto, esta redução torna-se cada vez menos acentuada e o número de computações da métrica tem tendência a piorar à medida que o número de *pivots* aumenta.

Neste caso, o número de *pivots* escolhido, para cada base de dados sobre um determinado espaço métrico, foi o que apresentou a menor média do número de

computações da métrica.

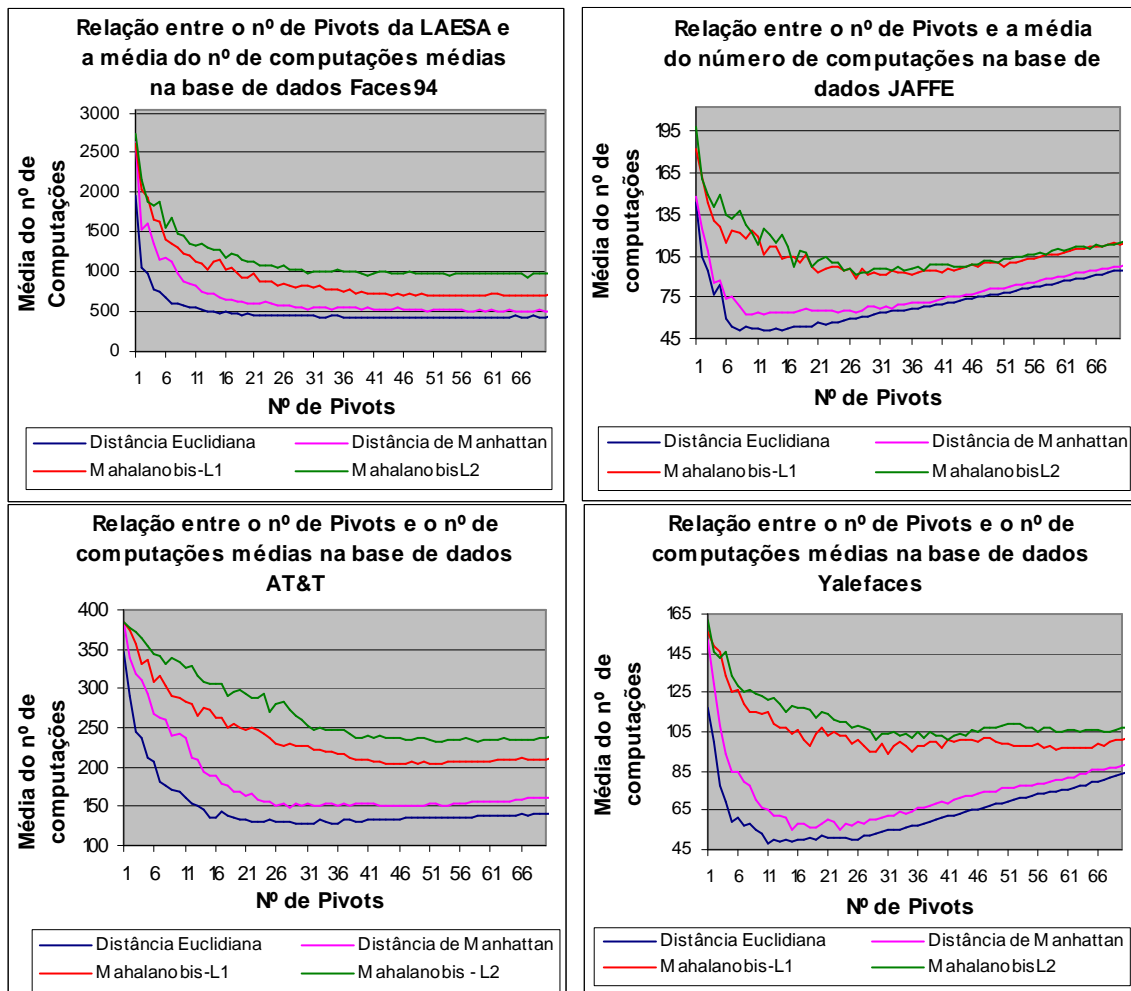


Figura A 3.1 – Parametrização da LAESA

### A 3.2. Parametrização da GNAT

Os gráficos da figura A3.2 apresentam o comportamento da *GNAT* quando da variação do número máximo de centros de cada nó em diferentes bases de dados e com as diferentes métricas.

Os gráficos demonstram que a média do número de computações diminui com o aumento do número máximo de centros por nó. Neste caso, a média do número de computações é mínima quando o número de centros do agrupamento é igual ao tamanho da base de dados. No entanto, a parametrização da *GNAT* com um número de centros máximo por nó igual ao tamanho da base de dados conduz a uma estrutura com um único nível. Nestas condições, a tabela de intervalos de cada nó teria uma grande dimensão.

Desta forma, para avaliação final as estruturas *GNAT* foram parametrizadas com um número de centros máximo igual a 3.

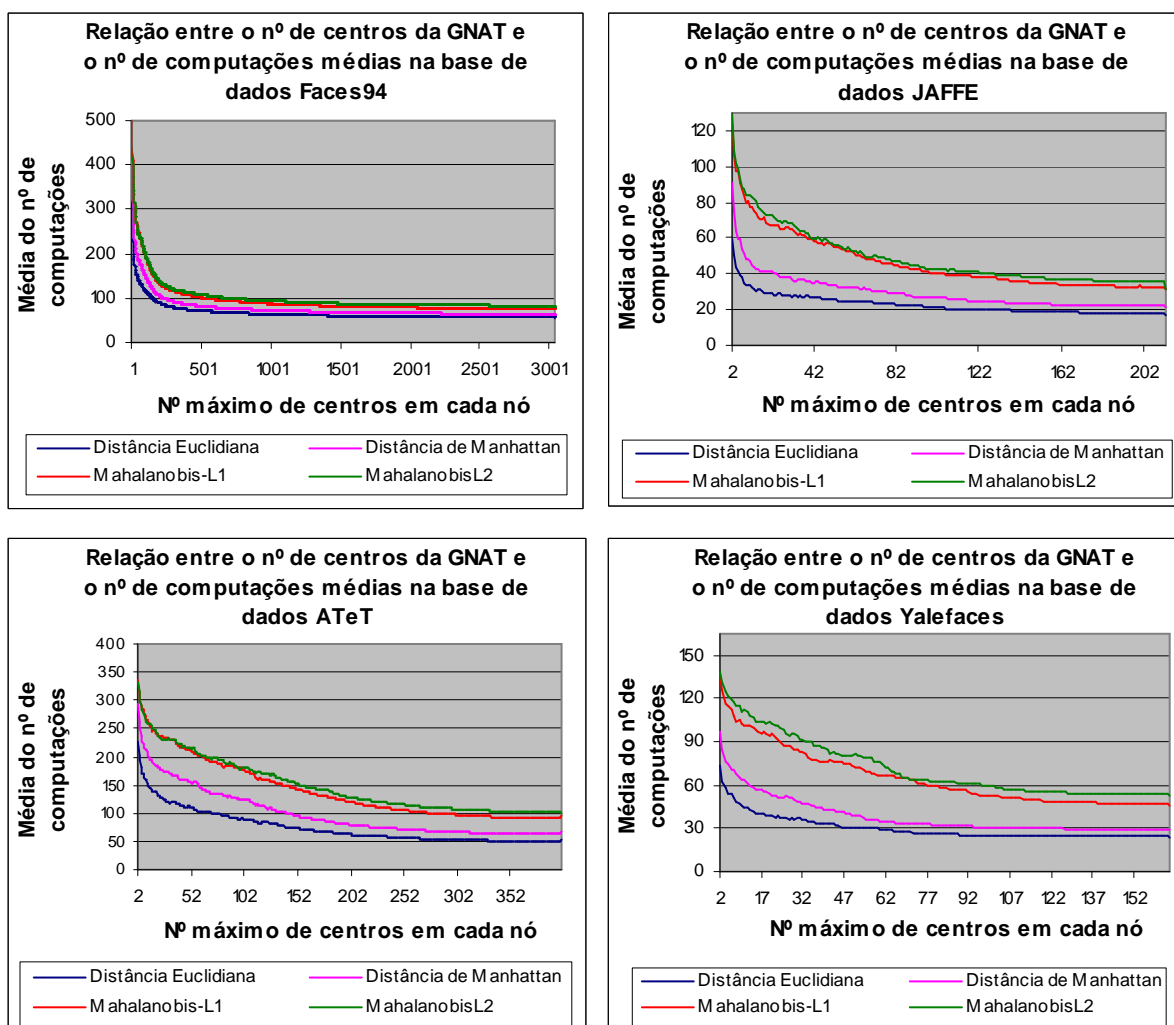


Figura A.3.2 – Comportamento da *GNAT* aquando da variação do nº de centros

### A 3.3. Parametrização da *DSAT* e variantes

A escolha da parametrização para estas estruturas foi realizada em conjunto. Desta forma, os gráficos das figuras A3.3, A3.4, A3.5 e A3.6 apresentam, respectivamente, o comportamento das várias estruturas com os diferentes espaços métricos nas bases de dados *Faces94*, *JAFFE*, *AT&T* e *Yalefaces*.

Os gráficos das figuras A3.3, A3.4, A3.5 e A3.6 permitem comparar a *DSAT* com as suas variantes. Neste caso, verificamos que, para o mesmo espaço métrico, se for utilizada a mesma aridade, na construção das diversas estruturas, a *DSAT* é a estrutura que tem sempre o pior desempenho em relação às outras duas e a *HDSAT2* é a que tem o melhor.

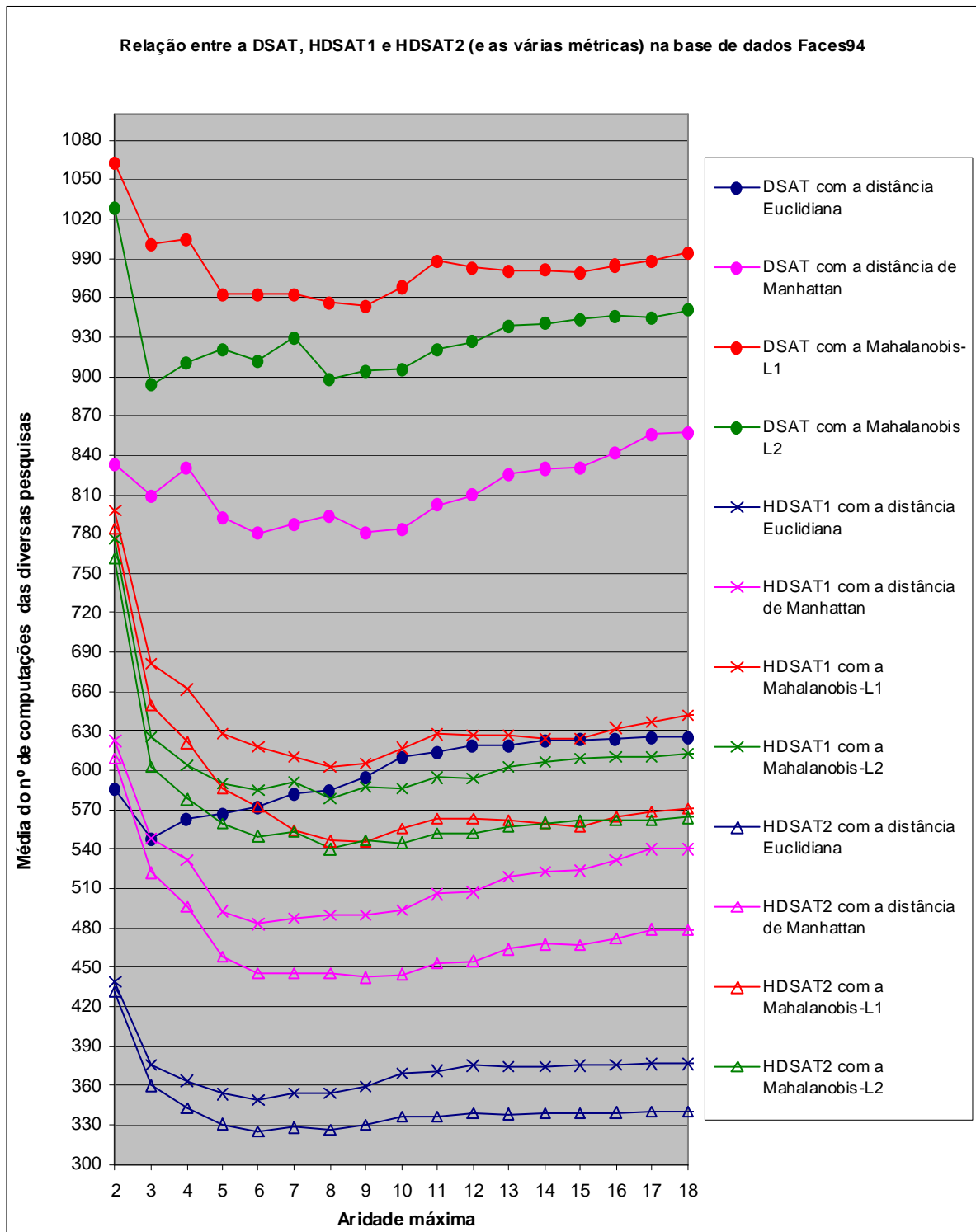


Figura A 3.3 – Parametrização da DSAT e variantes na Faces94

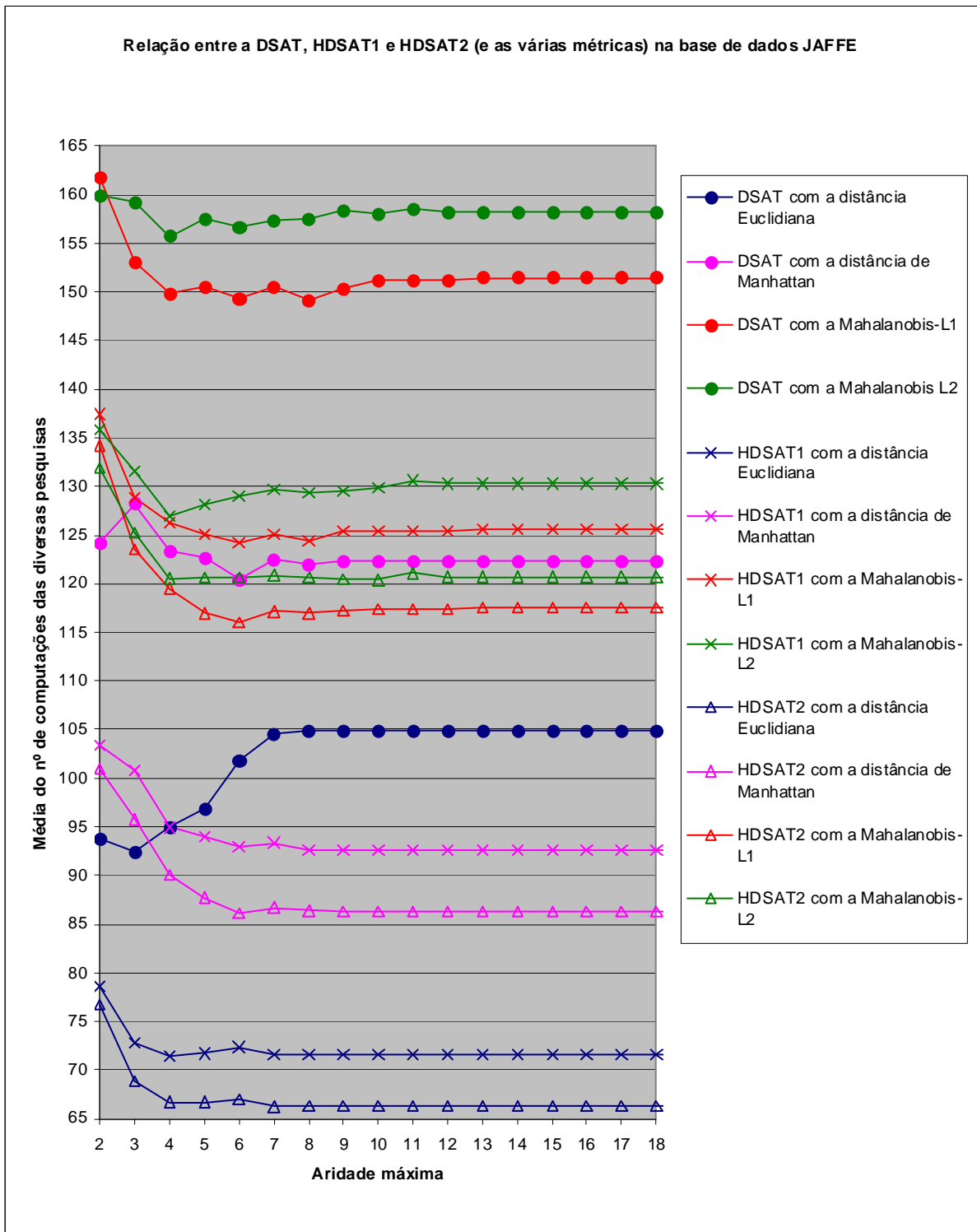


Figura A 3.4 – Parametrização da DSAT e variantes na JAFFE

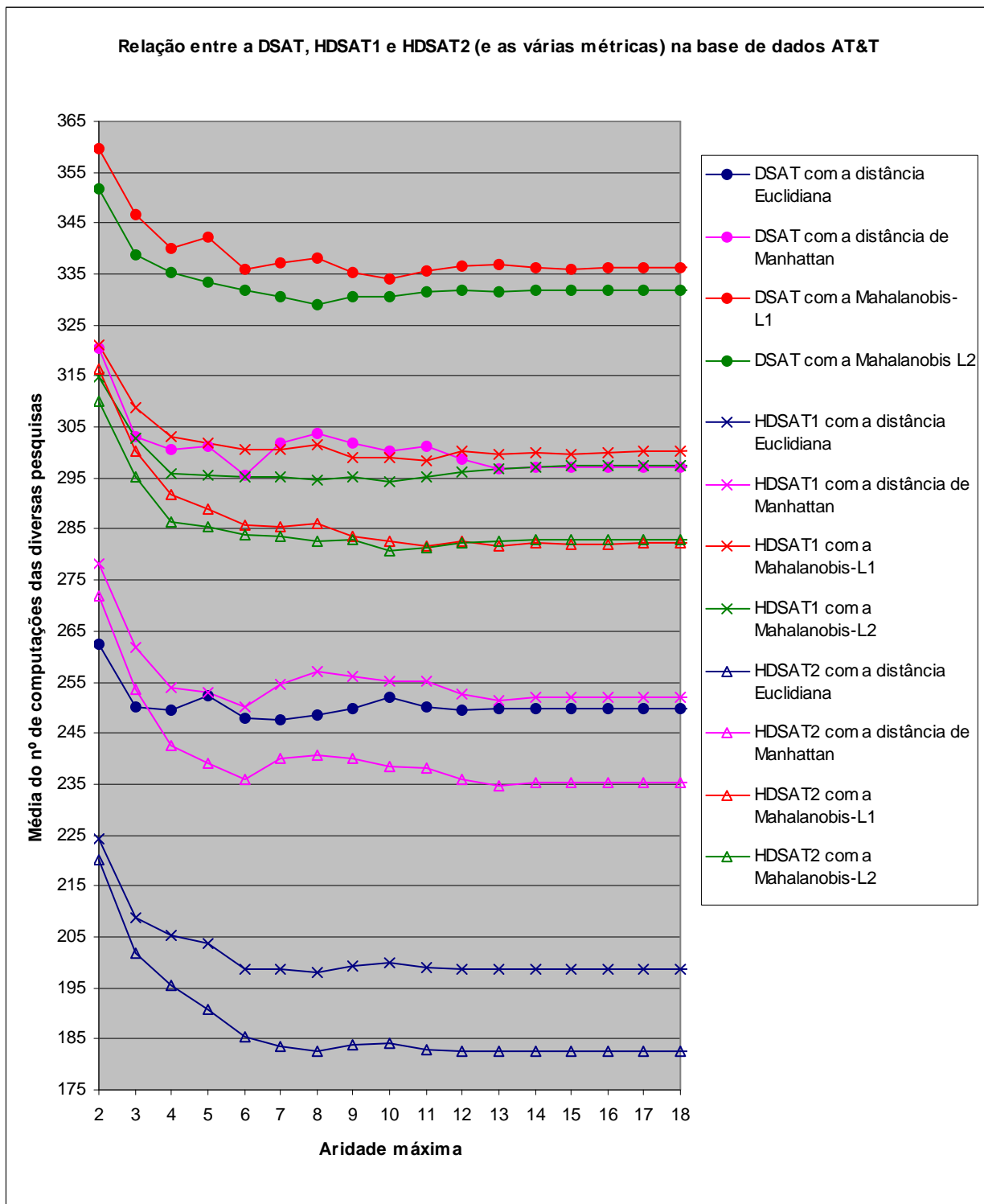


Figura A 3.5 – Parametrização da DSAT e variantes na AT&T

Relação entre a DSAT, HDSAT1 e HDSAT2 (e as várias métricas) na base de dados Yalefaces

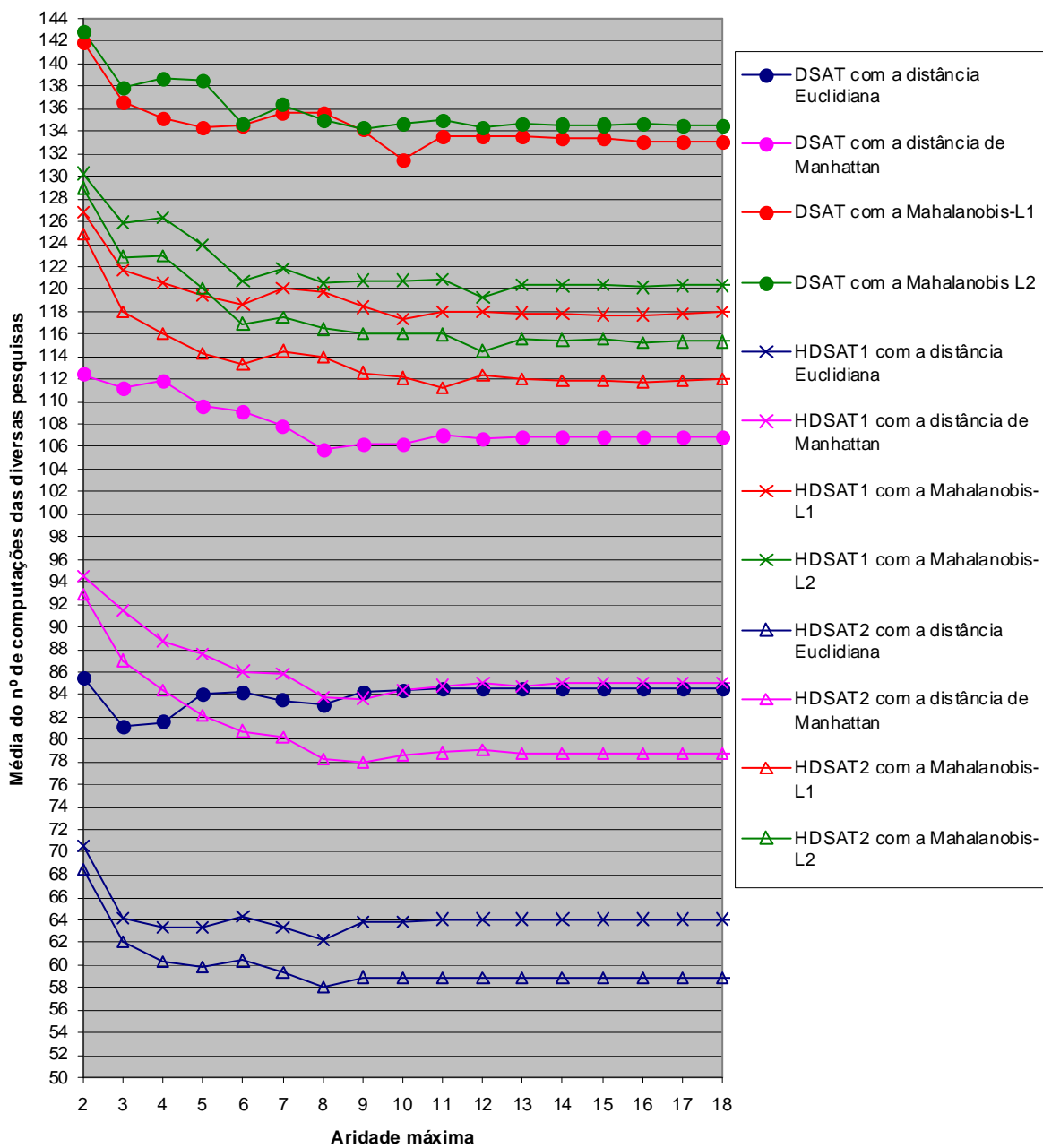


Figura A 3.6 – Parametrização da DSAT e variantes na Yalefaces

As árvores geradas pela *DSAT* e variantes têm a mesma estrutura, no entanto, com a *HDSAT1* e *HDSAT2* são utilizadas as informações dos *pivots* para descartar os vários elementos. É por esta razão que a *HDSAT1* e *HDSAT2* têm um desempenho superior à *DSAT*, no entanto, é necessário ter em conta que a *HDSAT1* e *HDSAT2* necessitam de mais espaço de armazenamento para guardar a informação das distâncias em relação aos vários *pivots*. Além disso, a *HDSAT2* tem um desempenho superior à *HDSAT1* pelo facto de usar mais *pivots* para descartar os vários elementos aquando de uma pesquisa.

Neste caso, a aridade máxima escolhida para cada estrutura numa base de dados sobre um determinado espaço métrico, foi a que apresentou a menor média do número de computações da métrica.

### A 3.4. Parametrização da *LC*

Os gráficos da figura A 3.7, A 3.8, A 3.9 e A 3.10 apresentam o comportamento da *LC* aquando da variação do parâmetro de construção da estrutura referente ao raio fixo de cada agrupamento, nas diferentes bases de dados com as diferentes métricas.

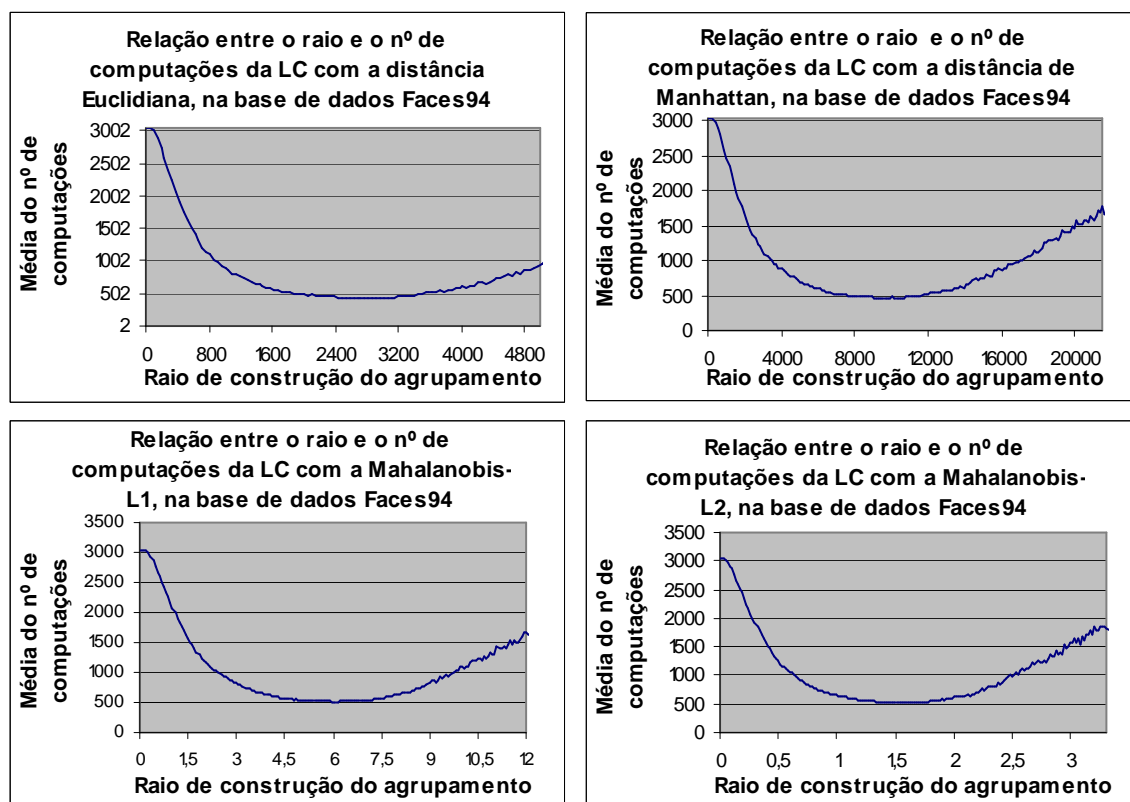


Figura A 3.7 – Comportamento da *LC* com agrupamentos de raio fixo na *Faces94*

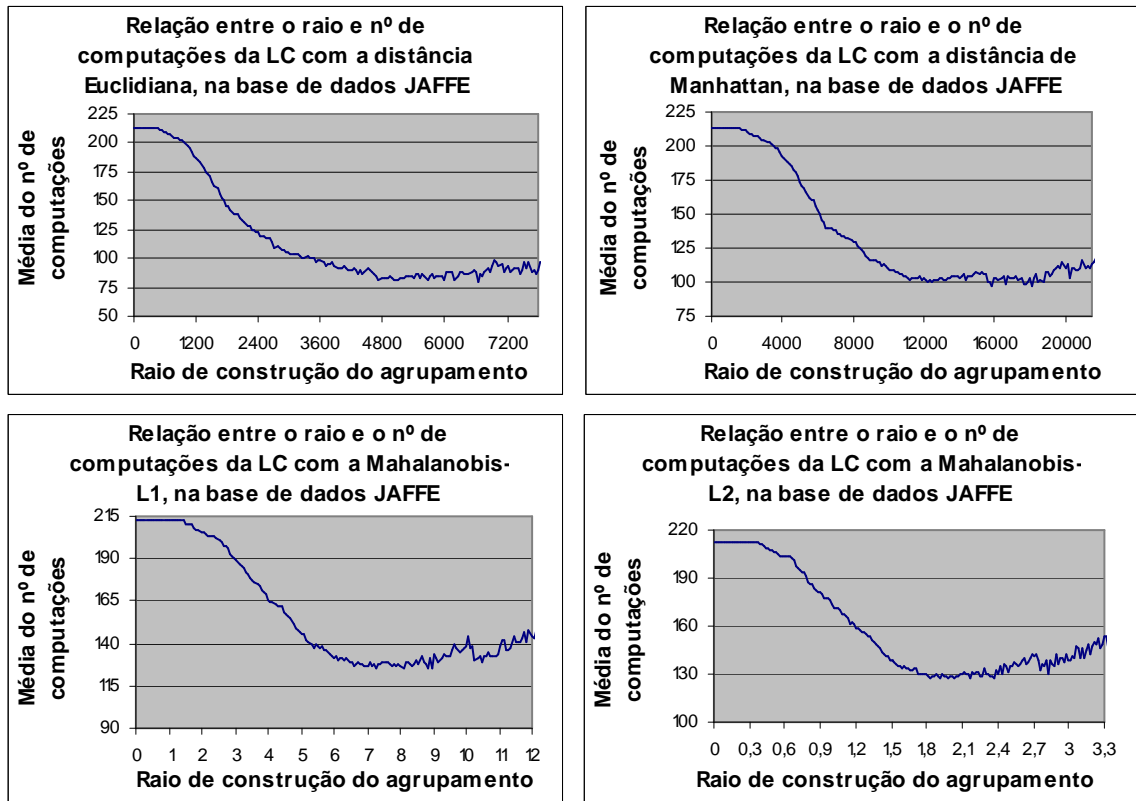


Figura A 3.8 – Comportamento da LC com agrupamentos de raio fixo na JAFFE

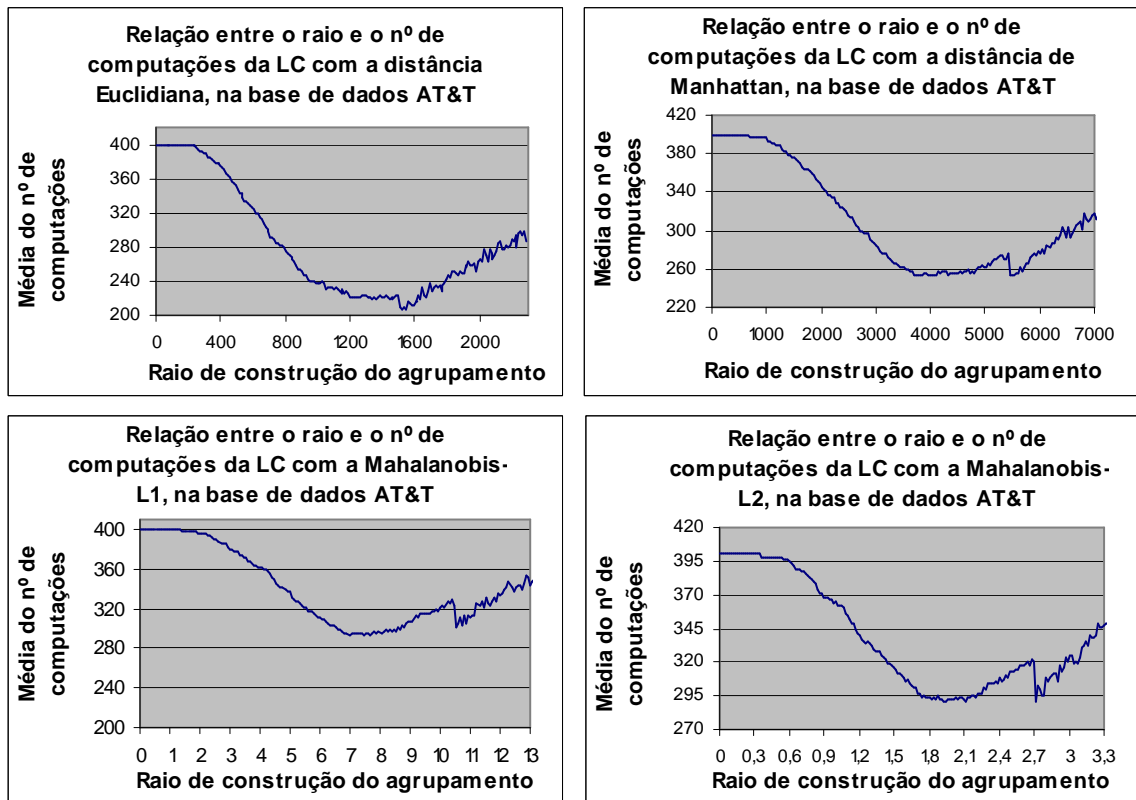


Figura A 3.9 – Comportamento da LC com agrupamentos de raio fixo na AT&T

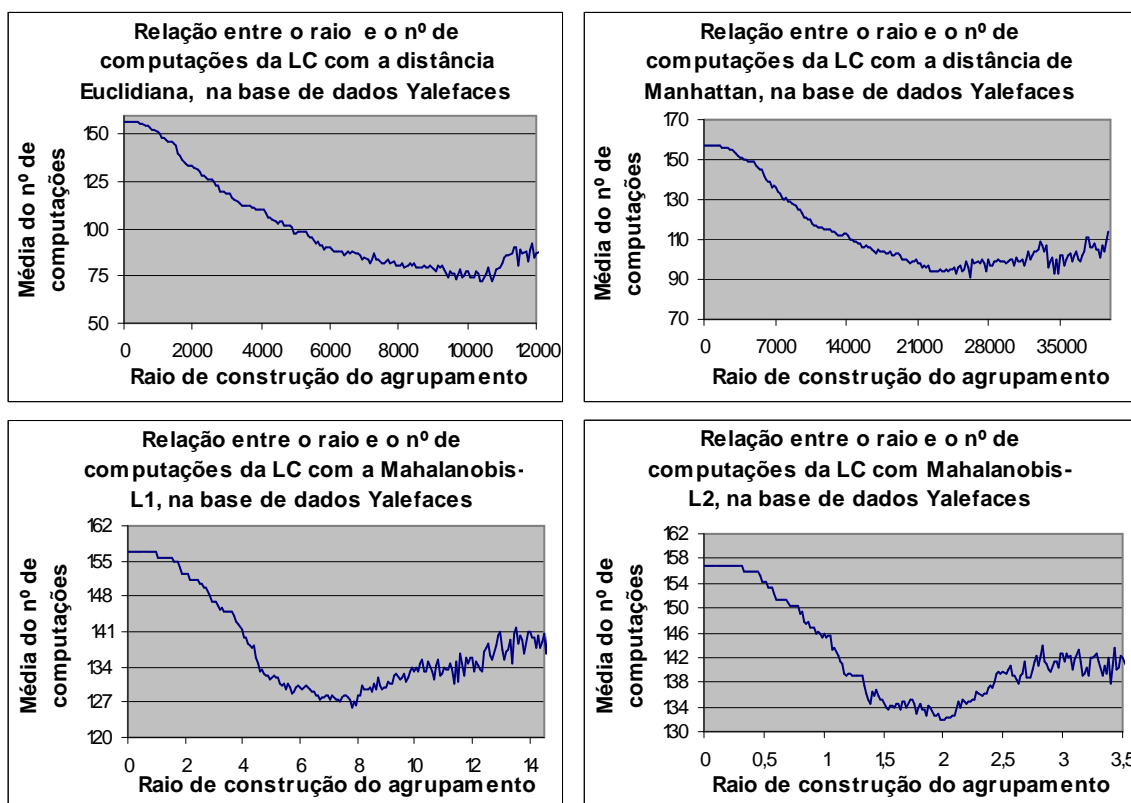


Figura A 3.10 – Comportamento da LC com agrupamentos de raio fixo na *Yalefaces*

Neste caso, o raio de cada agrupamento que foi escolhido, para cada base de dados sobre um determinado espaço métrico, foi o que apresentou a menor média do número de computações da métrica.

### A 3.5. Parametrização da *RLC*

O estudo do comportamento da *RLC* com os vários raios de construção dos agrupamentos foi realizado da seguinte forma: Para cada raio de construção foram calculadas as médias do número de computações correspondentes à parametrização da *RLC* com diferentes capacidades do vector. Neste caso, o intervalo de valores para a capacidade que foram testados está compreendido entre 2 e 26, inclusive. Desta forma, para cada raio é realizada a média das várias médias obtidas do nº de computações da *RLC* parametrizada com diferentes capacidades (associadas ao mesmo raio de construção). Em relação ao parâmetro do raio de construção, foi escolhido, para cada métrica, o raio com a menor média nas diferentes bases de dados. No que diz respeito ao parâmetro da capacidade de cada vector da *RLC* foi seleccionada a capacidade que obteve a menor média do número de computações da *RLC* (parametrizada com o raio escolhido anteriormente).

Nas figuras A3.11, A3.12, A3.13 e A3.14 é apresentado, respectivamente, o comportamento da RLC nas bases de dados *Faces94*, *JAFFE*, *AT&T* e *Yalefaces*.

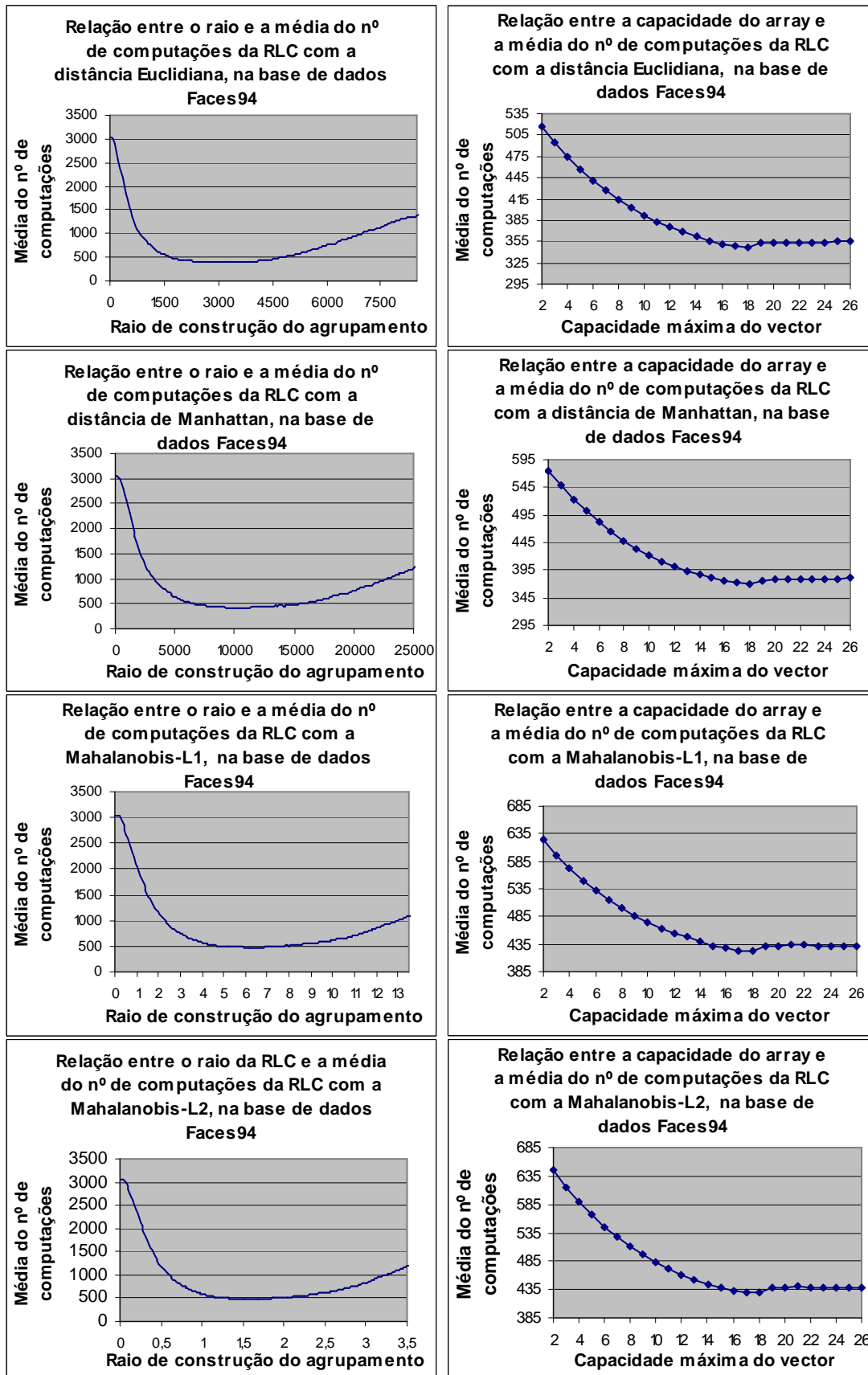


Figura A 3.11 – Comportamento da RLC na *Faces94*

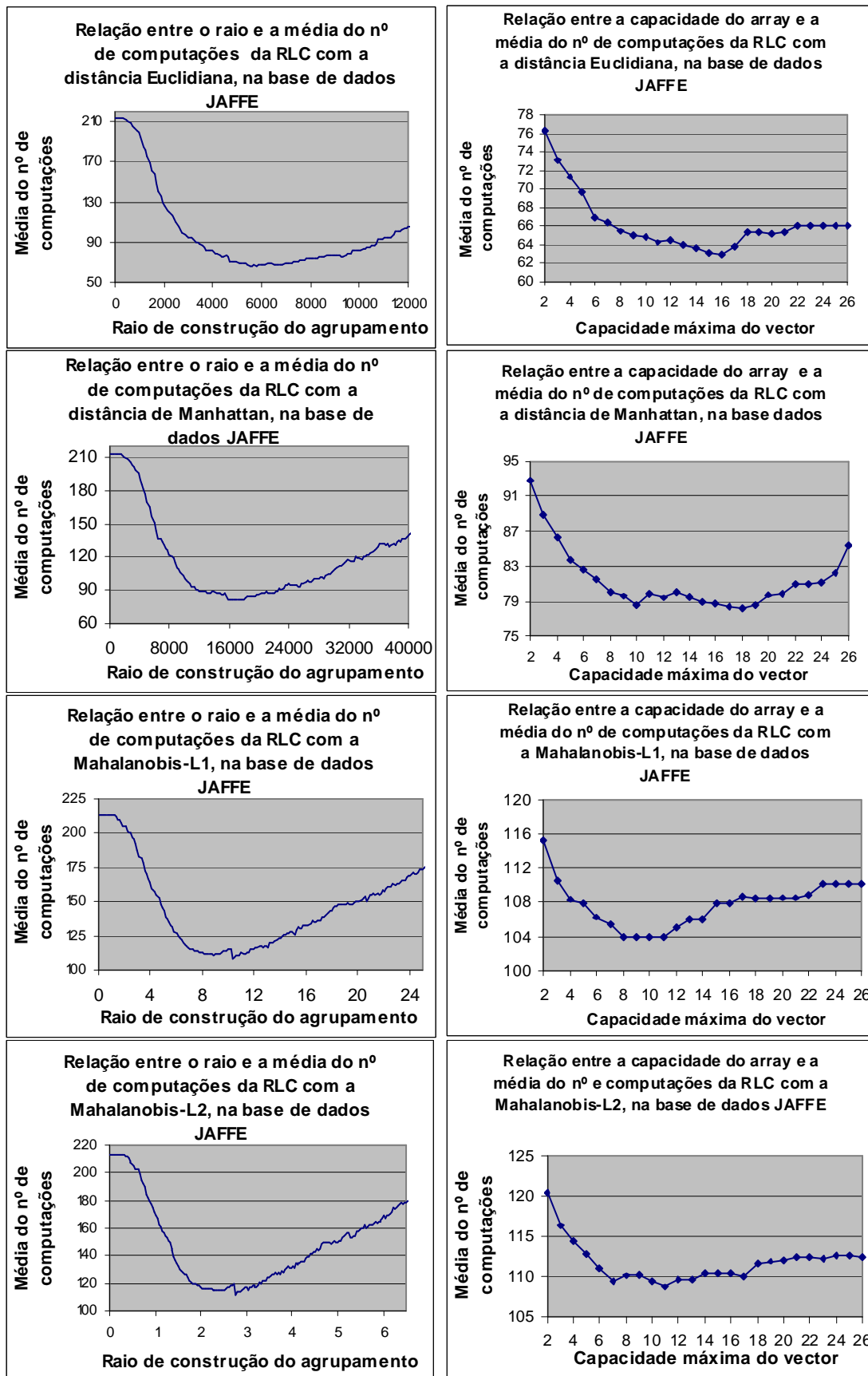


Figura A 3.12 – Comportamento da RLC na JAFFE

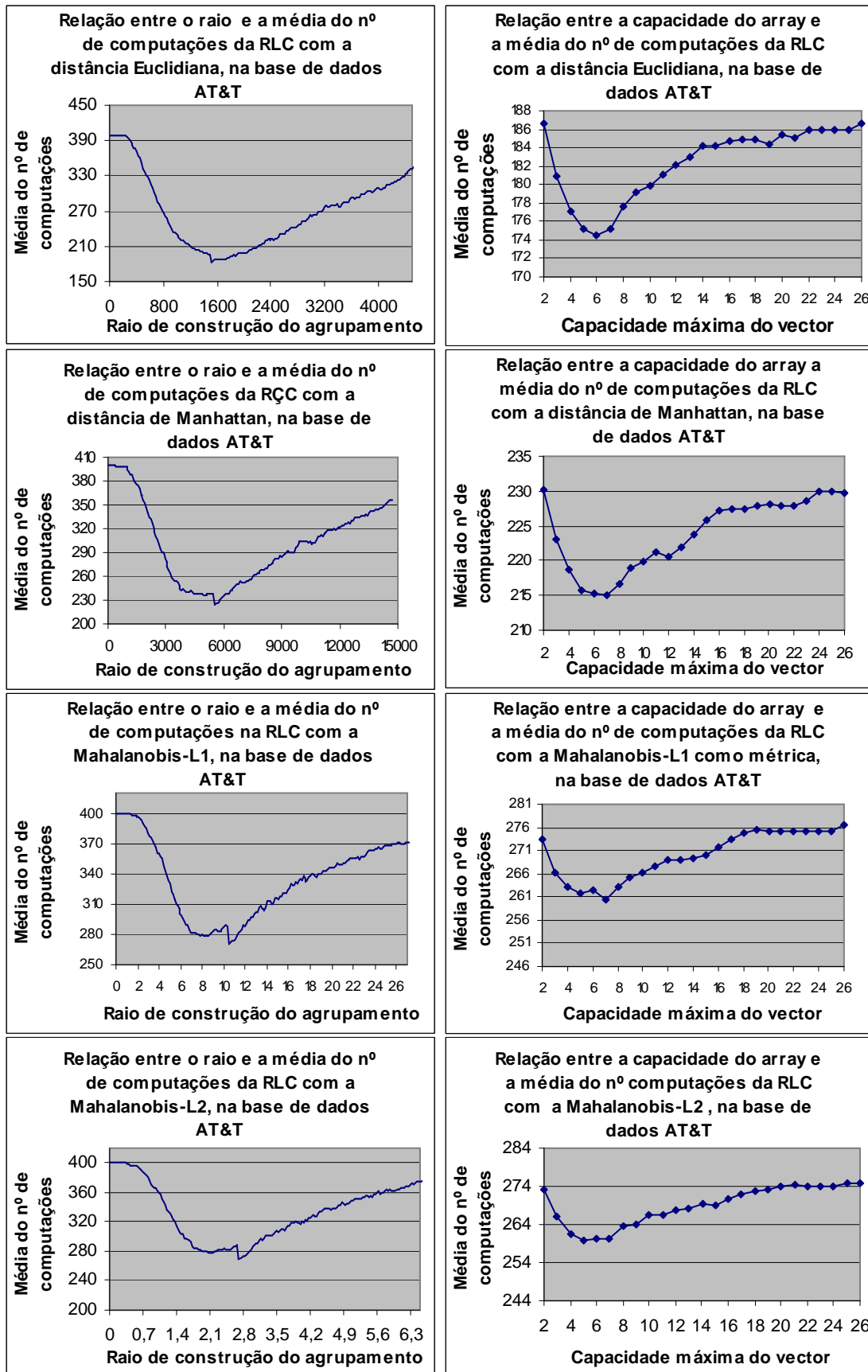


Figura A 3.13 – Comportamento da RLC na AT&T

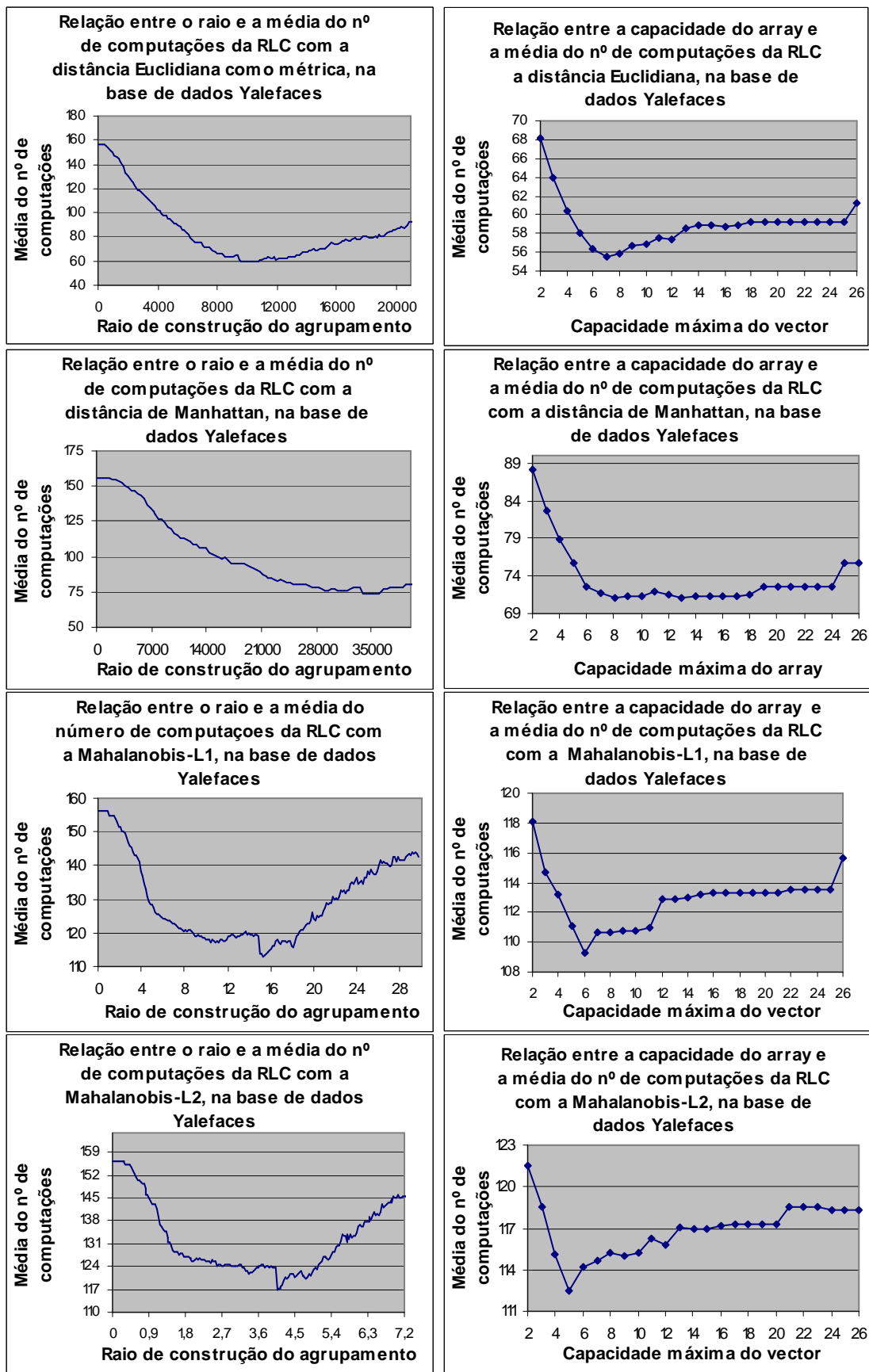


Figura A 3.14 – Comportamento da RLC na Yalefaces

Nestas figuras, cada gráfico na coluna do lado esquerdo relaciona o raio de construção da *RLC* com a média das várias médias obtidas do n° de computações da estrutura referentes à parametrização das estruturas com diferentes capacidades (com o mesmo raio de construção).

Cada gráfico da coluna da direita mostra o comportamento da *RLC* quando da variação do parâmetro que determina a capacidade de cada vector da *RLC*. Neste caso, foi utilizado o raio de construção que obteve a menor média do número de computações para as diferentes capacidades do vector.

Desta forma, o raio de construção do agrupamento escolhido, associado a uma base de dados sobre um determinado espaço métrico, foi o que apresentou a menor média do número de computações da métrica para as diferentes capacidades do vector. Em relação ao tamanho do vector, foi escolhido o que obteve a menor média do número de computações com esse raio.