RECUPERAÇÃO DE INFORMAÇÃO VISUAL EM BASES DE IMAGENS DE CIDADES HISTÓRICAS: CONTRIBUIÇÕES PARA A IDENTIFICAÇÃO E CLASSIFICAÇÃO DE IMAGENS

MARCELO DE MIRANDA COELHO

RECUPERAÇÃO DE INFORMAÇÃO VISUAL EM BASES DE IMAGENS DE CIDADES HISTÓRICAS: CONTRIBUIÇÕES PARA A IDENTIFICAÇÃO E CLASSIFICAÇÃO DE IMAGENS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Arnaldo de Albuquerque Araújo
Coorientador: Prof. Dr. Eduardo Valle

Belo Horizonte

Junho de 2013

Coelho, Marcelo de Miranda

Recuperação de Informação Visual em Bases de Imagens de Cidades Históricas: Contribuições para a Identificação e Classificação de Imagens / Marcelo de Miranda Coelho. — Belo Horizonte, 2013

xxxi, 127 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas Gerais
Orientador: Prof. Dr. Arnaldo de Albuquerque Araújo
Coorientador: Prof. Dr. Eduardo Valle

1. Clusterização. 2. Recuperação de Informação Visual. I. Título.

CDU

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Recuperação de informação visual em bases de imagens de cidades históricas: contribuições para o reconhecimento e classificação de imagens

MARCELO DE MIRANDA COELHO

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. ARNALDO DE ALBUQUERQUE ARAÚJO - Orientador
Departamento de Ciência da Computação - UFMG

PROF. EDUARDO ALVES DO VALLE JUNIOR - COORIENTADOR
Departamento de Engenharia da Computação e Automação - UNICAMP

PROF. CLODOVEU AUGUSTO DAVIS JÚNIOR
Departamento de Ciência da Computação - UFMG

PROF. DAVID MENOTTI GOMES
Departamento de Computação - UFOP

PROF. JACQUES WAINER
Departamento de Sistemas de Informação - UNICAMP

PROF. MARCOS ANDRÉ GONÇALVES
Departamento de Ciência da Computação - UFMG

PROF. NEUCIMAR JERÔNIMO LEITE
Instituto de Computação - UNICAMP

Belo Horizonte, 21 de junho de 2013.

À Nalu e Ana Laura, por me amarem incondicionalmente.
Aos meus pais, norteadores do meu caráter e, também, da minha curiosidade.

Agradecimentos

A Deus, princípio e origem de tudo, inclusive da ciência que ousamos tentar desvendar. A Ele meus agradecimentos, por continuar me amando mesmo nos momentos de afastamento extremo.

Pela aposta feita no meu sucesso, meus sinceros agradecimentos ao Prof. Arnaldo. Além disso, por ter me acolhido desde o nosso primeiro contato por volta do ano 2000 e ter pacientemente me aconselhado e me ajudado, ainda mais a partir de 2008, quando ingressei, definitivamente, como aluno de doutorado do DCC sob sua tutela.

Também por sua paciência, sou muito grato ao Prof. Eduardo. Nosso contato se iniciou, igualmente, nos anos 2000, sendo que eu era um recém mestre e ele um mestrando em início de curso. Anos depois, me vejo na situação de orientando do, então, recém doutor Prof. Eduardo. Vi essa possibilidade de interação com imenso prazer, pois já conhecia sua grande competência, mas a parceria foi muito além pois, além de coorientador, ele se mostrou um grande amigo.

Este trabalho não teria atingido o nível de qualidade atual se não fossem as observações sempre acertadas e enriquecedoras da banca, da qual fizeram parte os professores David Menotti Gomes, Marcos André Gonçalves, Clodoveu Augusto Davis Júnior e Neucimar Jerônimo Leite, que direcionaram o trabalho na defesa do projeto de tese e, juntamente como o Prof. Jacques Wainer, fizeram a avaliação final da tese. Aos senhores, meu muito obrigado.

Pelas fotografias da cidade de Ouro Preto que formaram o conjunto de imagens de consulta para os experimentos de identificação de cenas, meus agradecimentos ao Prof. Alexandre Leão, da Escola de Belas Artes da UFMG.

As contribuições do trabalho sobre a classificação de estilos arquitetônicos só foram possíveis devido à colaboração da arquiteta Branca Perocco, a quem sou muito grato pelo tempo dedicado a tirar minhas dúvidas e avaliar as anotações feitas nas imagens.

À equipe do VerLab, na pessoa do Prof. Mário Campos, meus agradecimentos pela colaboração durante todo este tempo e, principalmente, no desenvolvimento do suporte para fixação da Ladybug.

Como nenhuma pesquisa pode ser conduzida sem que haja recursos financeiros, as agências nacionais de fomento tiveram papel fundamental na realização dessa pesquisa. Agradeço principalmente ao CNPq, à CAPES, à FAPEMIG e à FAPESP.

Importante lembrar que este trabalho não poderia ter sido iniciado se não fosse o apoio dado pela Força Aérea Brasileira e pela Escola Preparatória de Cadetes do Ar (EPCAR). Sou muito grato às pessoas que, à época de minha liberação, participaram do processo: Tenente-Brigadeiro-do-Ar Antônio Pinto Macêdo, no Departamento de Ensino da Aeronáutica (DEPENS), e Major-Brigadeiro-do-Ar Alvani Adão da Silva, na EPCAR, assessorado pelo Coronel Domingos e pela Coronel Denise.

Não podia deixar de me lembrar, aqui, dos amigos do Núcleo de Processamento Digital de Imagens, os quais conheci durante a realização desse projeto. Pela acolhida inicial da Sandra e da Ana Paula, que me ajudaram nos meus primeiros passos no curso, tenho uma dívida enorme de gratidão. Na reta final tive a ajuda importantíssima de outros dois grandes amigos: Virgínia e Alberto — a ajuda de vocês foi crucial no período de conclusão do trabalho. Correndo o risco de deixar de citar alguém e já pedindo antecipadamente desculpas, quero agradecer também ao casal Rodrigo e Thatyene, Camilo, David Flam, Daniel Pacheco, Natália, Júlia, Fillipe, Tiago, Eduardo Gomes, Bruno, Guilherme.

Merece um destaque especial a ajuda do Cássio, com quem trabalhei diretamente como coorientador de IC e muito me ajudou no desenvolvimento deste trabalho, sendo ele o responsável pela parte penosa de alguns experimentos. A ele, a quem posso chamar também de amigo, meus sinceros agradecimentos.

Aos professores com quem convivi nesse tempo de UFMG e que contribuíram para minha formação além, é claro, do pessoal da administração, a quem agradeço imensamente, principalmente à Renata, Sheila, Sônia e Rosencler.

Aos amigos da EPCAR que sempre me direcionaram palavras de incentivo e de apoio, Vinícius, Alexandre, Vanderlan e tantos outros, obrigado. Também quero agradecer ao Charles, pelo apoio dado e, sem o qual, eu não poderia ter me afastado de minhas atividades para cursar o doutorado.

Uma casa construída sobre a areia não pode se sustentar e, da mesma forma, se não fossem as presenças das minhas amadas Nalu e Ana Laura em minha vida eu não teria tido forças para chegar até o fim da caminhada. À você, Nalu, que a duras penas renunciou a tudo para me acompanhar em mais esta empreitada, eu não tenho como expressar minha gratidão e nem como recompensar seu árduo sacrifício, mas espero

ser capaz de te demonstrar o quão importantes foram seu companheirismo e amor. À minha pequena Ana Laura, que também sofreu bastante nestes últimos anos, espero que encontre em mim o amigo com quem você sempre poderá contar durante a sua jornada.

Aos amigos e familiares que sempre acreditaram em mim e entenderam meus períodos de ausência, principalmente meus pais, Renato e Ilda, Frederico, Aline, Dedé, as Marias de Barbacena, Heitor, Betty, Alice, Tony e Liana. Nesses tempos de Belzonte, foram importantes também as presenças dos amigos Jário e Elinimar, além do apoio recebido do Tio Zé, Vilma, Marcos, Maria Inácia, Carlos Eduardo e Carminha.

Infelizmente, durante essa jornada, para muitas pessoas queridas eu não terei como recuperar o tempo perdido, mas nunca me esquecerei de vocês: D. Suely, Vô Chico, Seu Zé, Tio Agostinho e padre Afonso. Sei que vocês também torceram e continuam torcendo por mim.

"Mesmo que eu tivesse o dom da profecia,
e conhecesse todos os mistérios
e toda a ciência;
mesmo que tivesse toda a fé,
a ponto de transportar montanhas,
se não tiver caridade,
não sou nada."
(1º Coríntios, 13 2)

Resumo

Em meio a diversos desafios existentes na área de recuperação de informação visual em bases de dados de imagens, este trabalho procura contribuir em dois deles: a identificação de cenas e a classificação de imagens.

Em relação ao primeiro desafio, nos interessa a sua aplicação para a identificação de fachadas de edificações. Nela, descritores provenientes de obstáculos (como árvores, veículos e pedestres) e de elementos altamente texturizados das fachadas têm sido apontados na literatura como as principais causas para o insucesso da análise da semelhança entre cenas. Este trabalho oferece, então, uma abordagem de identificação de cenas em que os descritores escolhidos para representar as imagens são previamente filtrados, aumentando consideravelmente a quantidade de imagens corretamente identificadas. Uma importante característica dessa filtragem é o uso de algoritmos de clusterização em subespaço, capazes de atuar no espaço multidimensional ao qual pertencem os descritores de imagens.

Já para a classificação de imagens, nossa aplicação alvo é a diferenciação de estilos arquitetônicos de fachadas de edificações. Para isso, é apresentada uma extensão da técnica *Spatial Pyramids Matching* (SPM) de representação de imagens por dicionários visuais. A nova técnica, chamada de *Semantic Spatial Pyramids* (SSP), inclui a quantização dos descritores provenientes de regiões com significado semântico para a aplicação pretendida. Esta alteração tem se mostrado suficiente para superação da técnica tradicional, conforme os experimentos apresentados.

Adicionalmente, descrevemos uma solução de baixo custo para a aquisição de imagens *street-view* em cidades, visando a construção de bases de dados voltadas para as aplicações abordadas.

Palavras-chave: Reconhecimento de Cenas, Classificação de imagens, Aprendizagem não-supervisionada, Clusterização em Subespaço, Recuperação de Informação Visual.

Abstract

Researches on visual information retrieval applied to image datasets present plenty of challenges. Here, we intend to advance the state-of-the-art for two of them, namely scene recognition and image classification.

Regarding the first challenge, we focus on the application of recognizing building facades. Concerning this application, there are various works in literature which point out obstacles (e.g. trees, vehicles and pedestrians) and highly textured facade elements as the causes for unsuccessful similarity analysis between scenes. Thus, this work presents a new approach for scene recognition that previously filters the descriptors elected to represent the scenes. As consequence, we considerably increase the quantity of the scenes correctly recognized. An important aspect of the filtering process resides in the fact that it is made by subspace clustering algorithms. Such algorithms are capable of operating in the multidimensional subspace in which the image descriptors are inserted.

In respect to the image classification, the separation between architectural styles of building facades is the target application. Hence, we present an extension of the Spatial Pyramids Matching (SPM) method for image representation using visual dictionaries. This new technique, named Semantic Spatial Pyramids (SSP), introduces the descriptors quantization driven by regions semantically related to the desired application. Based on the experiments, we affirm that such a simple adjustment outperforms the traditional technique.

Additionally, we present a low-cost solution for street-view image acquisition in urban environments. This solution is used for generating datasets employed for the mentioned applications.

Keywords: Scene Recognition, Image Classification, Unsupervised Learning, Subspace Clustering, Visual Information Retrieval.

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos e Siglas

AKM	*Approximate K-means*
BoW	*Bag-Of-Words*
CBIR	*Content-Based Image Retrieval*
CE	*Clustering Error*
dod	*dimension-oriented distance*
E–MSSC	*Enhanced Mean-Shift for Subspace Clustering*
FINDIT	*Fast and INtelligent Subspace Clustering Algorithm using DImension VoTing*
FK	*Fisher kernel*
GMM	*Gaussian Mixture Models*
GPS	*Global Positioning System*
HKM	*Hierarchical K-means*
HTML	*HyperText Markup Language*
kNN	*k-Nearest Neighbors*
KNN–MRF	*K-Nearest-Neighbors–Markov Random Field*
LPP	*Locality Preserving Projections*
MAFIA	*Merging of Adaptive Finite Intervals*
MEC	*Maximum-Entropy Clustering*
MRF	*Markov Random Field*

MSSC	*Mean-Shift for Subspace Clustering*
NBNN	*Naïve-Bayes Nearest-Neighbor*
NN	*Nearest-Neighbors*
PB	*Probability of Boundary Edge Detector*
pLSA	*probabilistic Latent Semantic Analysis*
RAM	*Random Access Memory*
Rand	*Rand Index*
RANSAC	*RANdom SAmple Consensus*
RNIA	*Relative NonIntersecting Area*
SIFT	*Scale-Invariant Feature Transform*
SPM	*Spatial Pyramids Matching*
SSP	*Semantic Spatial Pyramids*
SURF	*Speeded-Up Robust Features*
SVM	*Support Vector Machine*
VI	*Variation of Information*
WFT	*Windowed Fourier Transform*

Glossário

A

Anotação de imagens
Associação de uma imagem a um determinado tema ou classe feita de forma manual por um observador [do inglês *Image annotation*].

C

Centroide
Aproximação matemática do representante de um conjunto de dados, que compartilham de certa similaridade. Geralmente, é obtido pelo valor médio dos pontos que representa.

Classificação de cenas
Classificação de uma imagem alvo em classes preestabelecidas [do inglês *Scene classification*].

Codebook
vide Dicionário visual.

Curvas de Preenchimento
Também chamadas de Curvas de Peano para o caso bidimensional, são curvas cuja extensão contém um quadrado bidimensional unitário ou, de forma mais genérica, hipercubo n–dimensional [do inglês *Space-filling curves*].

D

Descrição de imagens
Representação de imagens a partir de suas características morfológicas realizada por algoritmos destinados a este fim, tais como SIFT, SURF, GIST, etc. [do inglês *Image description*].

Descritores globais

Representam uma imagem a partir de suas características gerais, como cores, texturas e formas.

Descritores locais

Representam uma imagem a partir da relação entre pontos específicos desta, chamados pontos de interesse, e sua vizinhança.

Dicionário visual

Amostra do espaço de descritores locais de um conjunto de imagens escolhida para quantificar as ocorrências de cada instância da amostra nesse conjunto de imagens [do inglês *Codebook*].

I

Identificação de cenas

Busca de uma imagem alvo dentro de uma base de dados de imagens, retornando as imagens similares a esta [do inglês *Scene retrieval*].

M

Medoide

Instância de um conjunto de dados responsável por representar um determinado grupo de dados, segundo critérios de similaridade.

R

Reconhecimento de cenas

vide Identificação de cenas.

Sumário

Capítulo 1

Introdução

As bases de imagens georreferenciadas, tomadas em nível de rua[1], revolucionaram a relação das pessoas com seu ambiente, possibilitando a realização de visitas virtuais com detalhes e interatividade jamais antes disponíveis. Serviços como o Google Street View, extensão do bem-sucedido Google Maps[2], trouxeram ao grande público essa facilidade, que hoje já se banalizou como serviço fundamental para as grandes aglomerações urbanas.

A existência dessas extensas bases também estimulou interessantes desafios de pesquisa [Nokia, 2009], que já estão prestes a se tornar produtos de mercado. Uma possibilidade é a identificação de cenas urbanas, com o objetivo de recuperar com exata precisão a cena, fachada ou monumento capturado pela câmera de um dispositivo móvel (por exemplo, um telefone celular). A Figura 1.1 exemplifica uma aplicação em que uma foto capturada por um celular dever ser comparada e identificada em meio a um conjunto de cenas previamente capturadas. Nesses casos, o posicionamento por meio de um receptor GPS fornece pistas importantes, mas não é suficiente para determinar exatamente a cena. A exatidão é crítica para certos serviços de grande interesse, como a obtenção de informações turísticas sobre um monumento, ou recuperação de comentários dos frequentadores de um restaurante ou loja.

Outra possibilidade, de interesse mais especializado, diz respeito à classificação das fachadas em certas categorias, por exemplo, entre estabelecimento comercial ou residencial, ou, no que toca mais especificamente este trabalho, entre diversos estilos arquitetônicos. Essa classificação automática interessa (indiretamente) ao turista casual, mas se revela de importância fundamental aos profissionais de planejamento urbano e de conservação do patrimônio, que podem utilizá-la para facilmente identi-

[1] chamadas em inglês de *street-view*, termo consagrado a que daremos preferência

[2] http://maps.google.com

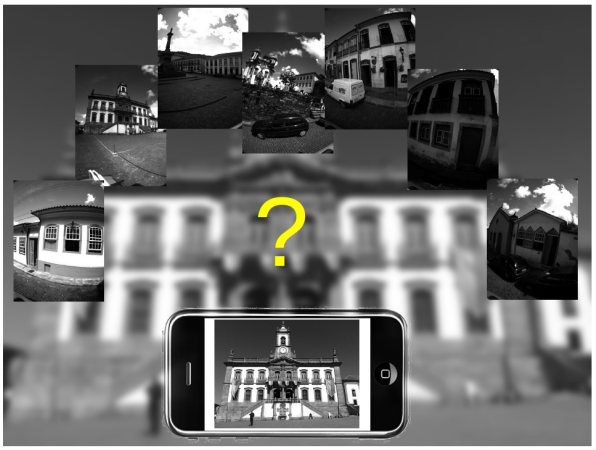


Figura 1.1: Problema da identificação de cenas.

ficar fachadas específicas ou até zonas inteiras da aglomeração urbana que devam ser alvo de atenção em políticas públicas, ou ações de preservação e restauro. A Figura 1.2 ilustra tal desafio, em que a cena em destaque dever ser associada a um dos dois grupos de cenas exibidas abaixo.

Figura 1.2: Problema da classificação de cenas.

Essas são, justamente, as duas aplicações-chave exploradas neste trabalho: o reconhecimento ou identificação de cenas, em que buscamos uma imagem específica numa base de fachadas; e a classificação de cenas, para a qual queremos determinar a categoria arquitetônica a que pertencem as imagens de uma base.

Em ambos os casos, concentramo-nos em cidades de Minas Gerais identificadas como importantes sítios de patrimônio histórico nacional e mundial. Consideramos que tanto a rica oferta de serviços turísticos digitais para essas cidades, quanto a criação de

soluções que favoreçam seu estudo e preservação são importantes questões de pesquisa, ainda não adequadamente resolvidas. Este trabalho é um passo em direção a avançar essas soluções.

Os serviços de busca e classificação em grandes bases de imagens, incluindo as de fachadas urbanas, demandam a criação de soluções automatizadas, pois o tratamento manual desses dados (visando por exemplo, anotação ou descrição) é inviável dado o seu volume, que, aliás, não para de crescer. Além das enormes bases encontradas em serviços como o Google Street View, existem hoje gigantescos repositórios criados por usuários comuns da Web — através de redes sociais como o Flickr[3], o Instagram[4], e o Facebook[5]. A taxa de postagem de fotos nessas redes é assombrosa, chegando a centenas de milhões por mês, no caso do Facebook[6]. Uma fração não negligível dessas imagens envolve fachadas, edifícios, monumentos, e outras que se enquadram como *street-view*, pois a presença de receptores de GPS em equipamentos ordinários disponíveis para o consumidor (celulares, câmeras compactas) banalizou o georreferenciamento das imagens. Obviamente, o tratamento manual desse fluxo de postagens é inexequível por razões de custo e rapidez mas, além dessas questões, o tratamento automático oferece promessas quanto aos espinhosos problemas da subjetividade e das inconsistências, que afetam o processamento por operadores humanos.

Desde o início da década de 1990, técnicas de recuperação de informação e classificação baseadas no conteúdo visual, que dispensam a presença de anotações, foram propostas [Bimbo, 1999]. Essas técnicas pioneiras baseavam-se na extração de características globais a partir de algum atributo da imagem como cor, textura ou forma, e frequentemente sintetizadas em um histograma. Entretanto, essa caracterização global apresenta uma série de deficiências, como o baixo poder de discriminação e a pouca robustez às condições de aquisição da imagem (iluminação, objeto de interesse em fundos complexos, etc.).

Nesse sentido, a primeira década do século XXI assistiu a duas sucessivas revoluções: a primeira com a ampla adoção pela comunidade de recuperação de informação multimídia dos descritores locais inspirados na visão por computador [Mikolajczyk & Schmid, 2001]; a segunda com a introdução de representações baseadas na agregação desses descritores locais, após um passo de quantização possibilitado pelo uso de um *codebook*, que se tornaram conhecidas como *bags of (visual) words* [Sivic & Zisserman, 2003]. Este trabalho se insere na continuidade dessas revoluções.

[3]http://www.flickr.com
[4]http://www.instagram.com
[5]http://www.facebook.com
[6]http://royal.pingdom.com/2013/01/16/internet-2012-in-numbers/

1.1 Contexto e Motivação

1.1.1 Recuperação e Classificação de Dados Visuais

A identificação e classificação de imagens baseadas no conteúdo, sem uso de anotações manuais ou outras pistas textuais, é um dos principais desafios da visão por computador. A percepção e interpretação humanas das imagens são difíceis de traduzir nos modelos matemáticos e estatísticos atualmente disponíveis. A extração de características de baixo nível, como cor, textura, e forma — e mesmo a extração de características mais sofisticadas, com uso de descrições locais e *bags of words*, são insuficientes para modelar a riqueza de interpretações semânticas que um ser humano pode associar às imagens. A essa distância entre a percepção humana e a representação computacional, dá-se o nome de *gap* ou abismo semântico.

Entretanto, apesar de os problemas de identificação e classificação para um conjunto genérico de imagens serem questões de pesquisa em aberto, soluções promissoras têm sido encontradas para contextos e aplicações mais específicos, em que já é possível obter acurácias de identificação e classificação aceitáveis.

Muitas propostas focam na identificação e classificação de imagens pertencentes a grandes bases de dados e que são distribuídas com suas anotações para permitir a comparação dos resultados. Por exemplo, as bases Caltech e suas variações[7], Pascal VOC[8] e LabelMe[9] são constantemente avaliadas na literatura [Lazebnik et al., 2006; Oliva & Torralba, 2001; Boureau et al., 2010; Perronnin et al., 2010; Avila et al., 2011]. Essas bases de dados são compostas de um certo número de classes de imagens e a taxa de acerto para as duas tarefas de recuperação de informação visual abordadas neste trabalho varia consideravelmente, evidenciando a necessidade de soluções guiadas pelo contexto.

Outra interessante vertente é a de reconhecimento de ações humanas em bases de vídeo como Hollywood[10] e Hollywood–2[11], que são usadas por Laptev & Perez [2007], Laptev et al. [2008], Marszalek et al. [2009] e Lopes et al. [2009, 2011] com o objetivo de se detectar automaticamente ações humanas como beber, sentar, entrar, sair, etc. Novamente, percebe-se que a aplicação para contextos específicos tem sido explorada na literatura.

[7] http://www.vision.caltech.edu/archive.html
[8] http://pascallin.ecs.soton.ac.uk/challenges/VOC/databases.html
[9] http://labelme.csail.mit.edu/
[10] http://lear.inrialpes.fr/people/marszalek/data/hoha/hollywood.tar.gz
[11] http://www.irisa.fr/vista/actions/hollywood2

Tanto a identificação de cenas quanto a classificação de imagens apresentam diversos problemas em aberto, entre eles a grande sobrecarga de ocupação de memória e processamento computacional gerados pela avaliação de milhões de descritores para a comparação ou casamento entre as cenas, bem como a existência de descritores pouco discriminantes que interferem no sucesso do processo de identificação das cenas. Além disso, há o desafio de se tratar detalhes semânticos presentes nas imagens para que seja realizada uma correta classificação das mesmas. As bases de dados citadas anteriormente desafiam a comunidade científica na resolução de diversos problemas de interesse geral. Entretanto, para questões mais específicas como a recuperação de informação visual utilizando cenas de cidades históricas esbarra-se, ainda, na falta de bases de dados devidamente anotadas.

1.1.2 Projeto Cidade Virtual e Cidades Históricas de Minas

De 2008 a 2010, o NPDI — Núcleo de Processamento Digital de Imagens[12] desenvolveu o projeto Cidade Virtual. O projeto, para o qual o desenvolvimento desta tese colaborou, ambicionou a criação de uma metodologia e meios tecnológicos para aquisição de bases do tipo *street-view*, com custo reduzido, mas qualidade suficiente para fomentar aplicações digitais de turismo e conservação do patrimônio. O projeto contemplou a coleta das imagens de várias cidades históricas mineiras, a construção de uma interface de navegação virtual, e, no que toca especificamente este trabalho, a proposta das técnicas de aquisição das imagens, e, principalmente, as aplicações relacionadas à recuperação e classificação baseadas no conteúdo já citadas.

Os vídeos capturados para o projeto se tornaram insumo para interessantes linhas de pesquisa, tanto as relacionadas à Ciência da Computação, quanto as de outras áreas do conhecimento, como Arquitetura e Belas Artes. Os vídeos também podem ser coadjuvantes em esforços de preservação do patrimônio cultural nessas cidades.

A identificação de fachadas e monumentos em imagens de *street-view* tem recebido muita atenção das comunidades de Visão por Computador e Recuperação de Informação Multimídia, chegando a ser proposta como um Grande Desafio na conferência ACM Multimedia de 2009[13]. Muitas soluções envolvem a descrição (usualmente com características locais) das imagens da base e da imagem-consulta, e uma busca por similaridade dos descritores da imagem-consulta nos descritores da base[Lowe, 2004; Picard et al., 2009; Valle et al., 2009; de M. Coelho et al., 2011]. Uma vez efetuada a

[12] Laboratório vinculado ao Departamento de Ciência da Computação da Universidade Federal de Minas Gerais.

[13] *Where was this photo taken, and how?*[Nokia, 2009]

busca por similaridade, sistemas de contagem de votos, assistidos ou não por mecanismos mais sofisticados de remoção de emparelhamentos espúrios, permitem encontrar a imagem da base que melhor se ajusta à imagem-consulta.

A classificação em estilos arquitetônicos das fachadas que tenham elementos de estilo em comum são temas de interesse mais especializado, mas que encontram eco em trabalhos recentes da literatura [Mathias et al., 2011; Shalunts et al., 2011, 2012a,b]. Neste trabalho, estamos, em particular, interessados em classificar a arquitetura barroca colonial de Minas Gerais, de estilo comumente chamado "Barroco Mineiro". Embora esse estilo tenha certos elementos que o caracterizam, fazendo com que um observador humano consiga com certa facilidade distinguir uma fachada de época perfeitamente preservada de uma fachada contemporânea, esses elementos apresentam, também, bastante diversidade visual, dificultando a classificação automática (Figura 1.3). Queremos, ainda, evitar o uso de regras *ad-hoc*, que limitam a aplicabilidade de certos trabalhos encontrados na literatura [Shalunts et al., 2012a]: ao contrário, almejamos um arcabouço flexível, que possa ser adaptado a diferentes necessidades futuras.

1.2 Objetivos e Hipóteses Científicas

Nosso principal objetivo é avançar o estado da arte da identificação e classificação automática de imagens. Em ambos os casos, elegemos como aplicações-chave a identificação de cenas e a classificação de estilos arquitetônicos em bases de imagens tomadas em *street-view*.

Colimando ainda mais o escopo, incluímos esse trabalho no contexto da aplicação da Tecnologia da Informação à conservação do patrimônio cultural, escolhendo cidades históricas de Minas Gerais para adquirir as imagens de *street-view*, e testando nossas técnicas nessas bases. Entretanto, ressaltamos que as técnicas também têm potencial aplicação em um escopo ampliado. A escolha da aplicação de conservação do patrimônio no contexto brasileiro levou ao objetivo de se criar metodologias e meios técnicos para aquisição das imagens de *street-view* com boa qualidade e baixo custo.

Os descritores locais, citados anteriormente, têm larga utilização na literatura e são fundamentais no processo de identificação de cenas. Entretanto, para imagens do tipo *street-view*, alguns autores [Valle et al., 2009; Picard et al., 2009; Turcot & Lowe, 2009] identificaram um problema trazido pela detecção de grandes volumes de descritores pouco discriminativos, especialmente em zonas fortemente texturizadas das imagens, como a vegetação ou as sombras projetadas por ela. A identificação *a priori*

Figura 1.3: Exemplos visuais de elementos arquitetônicos encontrados nas construções da cidade de Ouro Preto–MG.

dessas regiões é, entretanto, problemática, de forma que a filtragem dos descritores pouco discriminativos foi identificada como uma possibilidade promissora [Turcot & Lowe, 2009]. Realizar essa filtragem, através de uma abordagem não-supervisionada (sem necessidade de aprendizado prévio) é um dos principais objetivos desse trabalho. Como consequência, tencionamos também avançar o estado da arte para clusterização dos descritores multimídia, que sofrem os problemáticos efeitos da alta dimensionalidade. Em particular, nos concentramos na família de métodos chamada de *clusterização em subespaço*, que explicamos mais detalhadamente na Seção 2.4, que suporta melhor os efeitos da dimensionalidade.

Quanto à classificação de estilos arquitetônicos, em nosso trabalho propusemos uma abordagem baseada no modelo de *bags of (visual) words*. Identificamos a possi-

bilidade de melhorar os resultados agregando mais informação espacial ao modelo de *bags*. Tradicionalmente para esses casos, é utilizado o modelo de pirâmides proposto por Lazebnik et al. [2006], com regiões fixas, não adaptativas aos dados. Nosso objetivo é avaliar o quanto o uso de regiões adaptadas às imagens (isto é, adaptadas aos elementos da fachada) pode melhorar os resultados de classificação. Embora nesse trabalho nos limitemos à aplicação de *street-view*, acreditamos que o uso de regiões adaptadas poderia ser útil para uma ampla gama de aplicações.

Elencamos, dessa forma, as seguintes hipóteses científicas a serem testadas no trabalho:

- A filtragem de descritores pouco discriminativos melhora os resultados da identificação de cenas em bases do tipo *street-view*;
- A filtragem de descritores pouco discriminativos pode ser feita com uma abordagem baseada em clusterização;
- O uso de regiões adaptadas a elementos semanticamente relevantes das fachadas (por exemplo, portas, janelas, telhados, etc.) melhora os resultados de classificação de estilo arquitetônico em bases do tipo *street-view*.

1.3 Contribuições

- Apresentação de uma metodologia de identificação de cenas na qual os descritores de imagens não discriminantes são filtrados e descritores com maior poder discriminativo são usados para o processo de identificação;
- Incremento na identificação de cenas através da filtragem não-supervisionada de descritores de imagens, utilizando os algoritmos de clusterização em subespaço FINDIT [Woo et al., 2004] e MSSC [Gan et al., 2007];
- Decorrente da metodologia de identificação de cenas com o emprego da filtragem não-supervisionada, aumento da eficácia do processo ao usar uma quantidade de descritores equivalente a 2% do volume original de descritores, proporcionando menor tempo de execução do algoritmo;
- Elaboração do algoritmo *Enhanced Mean-Shift for Subspace Clustering* (E–MSSC), que faz uso de cuidadosa amostragem da base de dados, ganhando muito em eficiência com perdas modestas de precisão;

- Apresentação de uma metodologia de classificação de imagens baseada em dicionários visuais e que emprega regiões semanticamente relevantes (adaptadas à aplicação) na construção das representações das imagens;

- Análise da robustez da metodologia ora proposta com a inserção de perturbações nas regiões usadas na construção das representações das imagens, evidenciando, mesmo assim, a superioridade da técnica em relação ao estado da arte;

- Comparação da metodologia proposta com o estado da arte na classificação de imagens por vocabulários visuais, apresentando, a primeira, as melhores taxas de classificação;

- Criação de uma metodologia e meios técnicos para aquisição de imagens do tipo *street-view* georreferenciadas com baixo custo, aplicável em cidades e comunidades brasileiras;

- Aquisição de bases de imagens para quatro cidades históricas de Minas Gerais: Ouro Preto, Congonhas do Campo, Tiradentes e São João del Rei;

- A partir das bases adquiridas, criação de conjuntos anotados de imagens: um para os testes de identificação de cenas, composto de 618 imagens da base de dados, 38 imagens de busca e sua verdade-terrestre[14]; e outro para a classificação de estilos arquitetônicos composto de 1000 imagens, devidamente anotadas no que tange à sua classificação de estilo e suas regiões semânticas (elementos de fachada), usadas para a construção das representações das imagens.

1.4 Publicações

- de M. Coelho, M.; Valle, E.; dos Santos Júnior, C.; Araújo, A. de A. (2011). Subspace clustering for information retrieval in urban scene databases. Em Proceedings of the XXIV Conference on Graphics, Patterns, and Images, SIBGRAPI '11, IEEE Computer Society,pp. 173–180.

- Lopes, A. P. B.; de Avila, S. E. F.; Peixoto, A. N. A.; Oliveira, R. S.; de M. Coelho, M.; Araújo, A. de A. (2009). Nude detection in video using bag-of-visual-features. Em Proceedings of the XXII Conference on Graphics, Patterns, and Images, SIBGRAPI '09, IEEE Computer Society, pp. 224–231.

[14] esse termo, do inglês *ground-truth*

- Valle, E.; de Avila, S.; da Luz Jr., A.; Souza, F.; de M. Coelho, M.; Araújo, A. de A. (2012). Content-based filtering for video sharing social networks. Em Proceedings of the Computational Forensics Workshop, XII Brazilian Symposium on Information and Computer System Security, SBSeg '12, Brazilian Computer Society, pp. 625–638.

- de M. Coelho, M.; Valle, E.; dos Santos Jr., C. E.; Araújo, A. de A. Identifying Street View Scenes through Unsupervised Feature Filtering. Pattern Analysis and Applications, Springer (em processo de revisão).

1.5 Organização do Texto

Para benefício da clareza, dividiremos o conteúdo desta tese em duas partes que abordarão, respectivamente, os problemas de identificação e classificação de cenas. No Capítulo 2, abordamos as questões fundamentais da representação de imagens e as técnicas empregadas na identificação de cenas. Ainda neste capítulo, são detalhados os algoritmos de clusterização empregados no avanço do estado da arte da identificação de cenas. Além disso, serão discutidos alguns dos principais métodos para identificação de cenas encontrados na literatura, juntamente com suas vantagens e dificuldades. As contribuições propostas para a identificação de cenas são vistas no Capítulo 3, destacando-se a filtragem dos descritores por algoritmos de clusterização e a extensão de um desses algoritmos, o algoritmo MSSC, no E–MSSC. O Capítulo 4 é reservado aos experimentos realizados para a identificação de cenas com o emprego da filtragem de descritores. Iniciando a Parte II, o Capítulo 5 discute as principais técnicas empregadas na classificação de imagens e, na sequência, os principais trabalhos da literatura que abordam a classificação de imagens. Em seguida, no Capítulo 6, apresentamos as contribuições propostas para o avanço da classificação de imagens empregando informação semântica e a extensão da representação piramidal usada em dicionários visuais para considerar regiões definidas por critérios semânticos. Os experimentos concernentes à classificação de imagens estão no Capítulo 7, sendo evidenciado, pelos resultados, o avanço obtido para a classificação de estilos arquitetônicos. No Capítulo 8, discutimos os ganhos obtidos com as contribuições propostas e os desafios remanescentes, sendo indicados trabalhos futuros. Há, ainda, dois apêndices voltados, respectivamente, para a metodologia e equipamentos utilizados na aquisição das bases de *street-view* (Apêndice A) e para as atividades extras desenvolvidas durante o trabalho (Apêndice B).

Parte I

Identificação de Cenas

Capítulo 2

Trabalhos Relacionados

Visando a identificação de cenas, algumas das principais técnicas relacionadas ao assunto são apresentadas e discutidas, com o objetivo de permitir um melhor entendimento do próximo capítulo que trata das contribuições propostas para o avanço do estado da arte da identificação de cenas.

Além disso, são abordados alguns dos principais trabalhos que envolvem o uso de descritores locais para a comparação de uma imagem alvo com as cenas de uma base, em busca de cenas similares à primeira. Admite-se, entretanto, a possibilidade de outros tipos de descritores como, por exemplo, aqueles que são extraídos no domínio da frequência e permitem a geração de uma assinatura semântica da imagem.

2.1 Descritores de Imagens

A extração de descritores de uma imagem é um processo crucial para os sistemas de Recuperação de Imagens Baseados em Conteúdo (*Content-Based Image Retrieval* (CBIR)) no qual cada imagem $\hat{I}$ é associada a um ou mais vetores $\vec{v} \in \mathbb{R}^n$, sendo $\vec{v}$ chamado de vetor de características da imagem $\hat{I}$ [da S. Torres & Falcão, 2006]. Os vetores produzidos são usados, então, para medir a similaridade entre as imagens, empregando-se, para isso, uma função de distância (por exemplo, Euclidiana) entre vetores de $\mathbb{R}^n$ [da S. Torres & Falcão, 2006; Penatti et al., 2012].

A mais importante divisão taxonômica dos descritores de imagens talvez seja entre os descritores globais e os locais. Descritores globais sumarizam informações da imagem como um todo — baseando-se em atributos como cor, textura e forma — em um único vetor de características. Descritores locais, por outro lado, extraem características de porções relativamente reduzidas da imagens: regiões salientes, bordas, ou pequenas áreas em torno de pontos de interesse, procurando descrevê-las através de

propriedades discriminantes e invariantes a transformações geométricas e fotométricas, frequentemente inspiradas na Visão por Computador [Tuytelaars & Mikolajczyk, 2008]. Os métodos usados para detecção dessas características locais se baseiam na curvatura do contorno dos objetos, na intensidade dos *pixels*, na distribuição de cor na imagem, em modelos de objetos, na invariância dos pontos em relação à sua vizinhança, na segmentação de objetos da imagem ou no aprendizado de máquina. Entretanto, os métodos mais populares são aqueles baseados na detecção de quinas (*corners*), uma vez que as quinas são detectáveis com mais segurança e precisão (em termos de invariância à localização) do que bordas ou regiões [Tuytelaars & Mikolajczyk, 2008]. Optamos, aqui, pelo uso do descritor *Scale-Invariant Feature Transform* (SIFT) por seu vasto uso na literatura da área de recuperação de informação visual.

2.1.1 *Scale–Invariant Feature Transform* (SIFT)

Este é um dos descritores mais comumente usados na literatura para a tarefa de identificação e classificação de imagens, por oferecer uma descrição extremamente distinta de pontos da imagem e invariante às mudanças afins de iluminação, à rotação e à escala [Lowe, 2004].

A primeira etapa do processo de descrição dos pontos consiste em identificar candidatos a pontos de interesse na imagem. Para isso, a imagem original é filtrada usando-se filtros Gaussianos de variância crescente e, em seguida, é calculada a diferença entre essas imagens, originando as diferenças-de-Gaussianas. O mesmo processo é repetido para versões reescaladas da imagem original. Posteriormente, cada ponto de uma determinada diferença-de-Gaussianas é comparado com seus oito vizinhos, na mesma imagem, e com seus nove vizinhos de cada uma das diferenças-de-Gaussiana adjacentes. São considerados candidatos aqueles pontos que se destacam entre os vizinhos por serem mínimos ou máximos locais.

Então, cada candidato é investigado com o intuito de se identificar quais são os pontos de interesse, ou seja, os pontos que serão descritos. São descartados os pontos de mínimo e máximo com baixo contraste e, também, aqueles que são respostas decorrentes de regiões de borda.

Os candidatos restantes são considerados pontos de interesse e para cada um deles é anotada sua posição na imagem, escala, ou a imagem suavizada à qual pertence, e a orientação do vetor gradiente. A orientação é observada através da presença de um pico no histograma de gradientes. Ocorrendo outros picos dentro da faixa de 80% do pico máximo, o mesmo ponto de interesse é anotado novamente, usando-se quantas orientações forem necessárias.

Finalmente, são gerados os descritores de cada ponto de interesse, acrescentando detalhes aos dados já anotados: localização do ponto na imagem, escala e direção do gradiente. Para cada ponto de interesse são calculados os módulos dos gradientes e suas orientações, em volta da localização do ponto, usando-se uma grade 4×4 e as 8 direções possíveis, o que produz um vetor de 128 dimensões. Além disso, é empregada a escala na qual o ponto foi selecionado, recuperando-se a função de borramento relativa ao momento da escolha do ponto. Uma função de peso Gaussiana é utilizada para dar a mesma chance para os pontos dentro da janela de pontos que têm o ponto de interesse como centro. Ao final do processo de construção do vetor, o mesmo é alterado para reduzir as interferências provenientes da iluminação e normalizado, para reduzir a influência de altos valores de gradiente.

Os descritores gerados são invariantes à escala e rotação, e robustos com relação à distorção gerada por transformações afins de iluminação e adição de ruído [Lowe, 2004].

A partir dos vetores de características, o próximo desafio para a tarefa de identificação de cenas é a comparação da similaridade entre vetores de duas cenas para inferir sobre a similaridade das próprias cenas. Torna-se necessário, portanto, o uso de alguma técnica de indexação dos descritores, haja vista a impraticabilidade da comparação direta entre os descritores. Descreveremos, a seguir, uma das técnicas possíveis, chamada de Multicurves [Valle et al., 2008].

2.2 Multicurves

Multicurves é uma técnica de indexação de descritores multimídia de alta dimensionalidade que, a partir de curvas de preenchimento, ordena os dados conforme sua vizinhança. O método consiste em realizar a busca por vizinhos mais próximos (*k-Nearest Neighbors* (kNN)) por meio da projeção dos pontos da base de dados e da imagem alvo em curvas de preenchimento do espaço criadas para conjuntos distintos de dimensões. A partir das projeções, são observados os pontos da base de dados próximos a cada ponto da imagem alvo dentro de uma mesma curva, possibilitando a escolha da imagem similar à imagem alvo como sendo a imagem da base de dados com o maior número de pontos próximos aos pontos da imagem de consulta (Figura 2.1) [Valle et al., 2008].

Segundo Valle et al. [2008], a técnica Multicurves é eficaz, tendo superado os métodos do estado da arte no índice de acertos, e eficiente, visto que consegue melhores resultados acessando um número reduzido de pontos em relação aos algoritmos usados

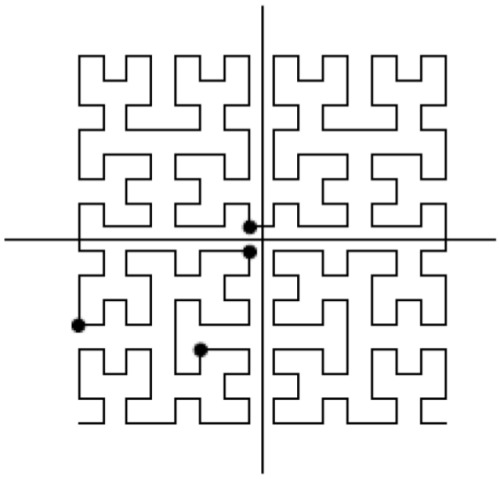


Figura 2.1: Exemplo extraído de Valle et al. [2008] que ilustra a ocorrência do problema das bordas na divisão do espaço em curvas, isto é, pontos próximos no espaço de descritores são projetados distantes na curva. A solução encontrada pelos autores foi verificar a proximidade em várias curvas que são criadas para subconjuntos específicos de dimensões.

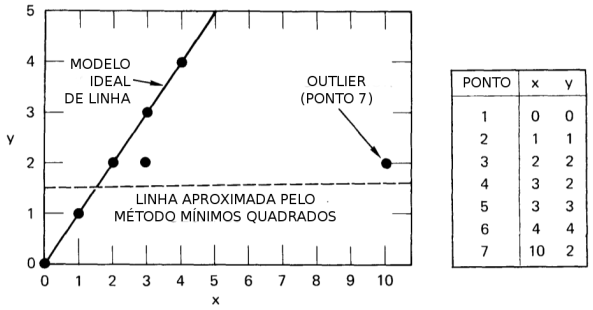
para comparação ou, na pior das hipóteses, a mesma quantidade de pontos que estes, além de ser adaptável ao uso de memória secundária.

Nem sempre a comparação direta dos vetores de características produzidos por um descritor de imagens é suficiente para garantir a eficácia da identificação de cenas. Dessa forma, pode ser empregado alguma outra técnica que refine a avaliação feita pelos vetores de características. Uma das técnicas usadas é a verificação da consistência geométrica entre os pontos de duas cenas, comparadas pelo método *RANdom SAmple Consensus* (RANSAC) que é comentado a seguir.

2.3 *RANdom SAmple Consensus* - RANSAC

O algoritmo *RANdom SAmple Consensus* (RANSAC) é uma proposta de solução para o problema de determinação da localização feita por Fischler & Bolles [1981]. Esse problema estabelece que a partir de um conjunto de marcações feitas em uma imagem, cujas posições são conhecidas, deve-se determinar a posição no espaço onde foi realizada a captura da imagem. A principal vantagem do algoritmo RANSAC em relação à solução comumente utilizada para esse problema, aproximação por mínimos quadrados, é sua robustez à ocorrência de pontos espúrios. A vulnerabilidade do método dos mínimos quadrados é evidenciada na sua aplicação em um conjunto de pontos onde a quase totalidade destes está alinhada e apenas um deles está bem distante dos outros.

Na tentativa de se ajustar um modelo a este cenário por meio dos mínimos quadrados, é obtida uma reta desalinhada com o conjunto principal de pontos, como pode ser visto na Figura 2.2.



PONTO	x	y
1	0	0
2	1	1
3	2	2
4	3	2
5	3	3
6	4	4
7	10	2

Figura 2.2: Exemplo extraído e adaptado de Fischler & Bolles [1981] que mostra a dificuldade do método mínimos quadrados de lidar com determinados conjuntos de dados.

Ao invés de realizar a aproximação do modelo baseado em todos os pontos, o método RANSAC procura ajustar um modelo aos dados, iterativamente, empregando o menor número possível de pontos, conforme a configuração dos parâmetros livres estabelecidos para o método. Dessa forma, elimina-se o problema dos pontos espúrios. Em seguida, esse grupo inicial de pontos é aumentado, desde que seja possível manter a consistência do modelo. O conjunto de pontos usado para aproximar o modelo é denominado conjunto de consenso. No exemplo da Figura 2.2, como o algoritmo não atingirá um consenso empregando os sete pontos para o modelo, o maior conjunto consensual possível será usado, alcançando a reta e desprezando o ponto espúrio na posição (10,2) [Fischler & Bolles, 1981].

Superadas as questões de avaliação da similaridade entre descritores e o refinamento geométrico dos descritores similares, ainda resta o problema de descritores não discriminantes relatado na literatura [Lowe, 2004; Valle et al., 2009; Picard et al., 2009]. Para solução desse desafio, pretendemos empregar a filtragem não supervisionada dos descritores por meio de algoritmos de clusterização, estudados na sequência.

2.4 Clusterização em Subespaço

O principal objetivo dos algoritmos de clusterização é dividir um conjunto de dados em grupos ou clusters a partir da análise da similaridade entre os dados. Percebe-se, então, a importância da escolha da função de similaridade usada, uma vez que, de acordo com a natureza dos dados, ela tem influência direta na formação dos clusters.

Entretanto, para dados com grande número de dimensões, os mesmos algoritmos de clusterização que atuam satisfatoriamente em baixas dimensionalidades são incapazes de garantir uma resposta eficaz. Isso ocorre devido à amplamente conhecida maldição da dimensionalidade (*curse of dimensionality*), termo criado por Bellman [1957]. Esse efeito implica na impossibilidade de se inferir a distribuição dos dados na presença de alta dimensionalidade, exigindo, para tal, um aumento exponencial da quantidade de amostras [Duda et al., 2001], uma vez que esses dados se tornam extremamente esparsos. Dessa forma, as distâncias máximas e mínimas entre os pontos convergem para um valor único [Parsons et al., 2004], o que dificulta a avaliação dos resultados produzidos pelas funções de similaridade. Por outro lado, dados reais tendem a ficar confinados em espaços cuja dimensionalidade efetiva é baixa, o que possibilita a resolução, mesmo para dados de alta dimensionalidade, de problemas como a clusterização [Bishop, 2006].

Devido a essa dificuldade em se trabalhar em alta dimensionalidade, nos dados provenientes da descrição de imagens através de algoritmos como *Scale-Invariant Feature Transform* (SIFT) [Lowe, 2004] ou *Speeded-Up Robust Features* (SURF) [Bay et al., 2006] o uso dos métodos clássicos de clusterização como o *K–means* não são indicados, pois eles são apropriados para bases com dados de baixa dimensionalidade. Isso ocorre porque os métodos clássicos tentam avaliar a similaridade dos dados em todas as dimensões existentes. Sendo assim, é possível que tais técnicas deixem de identificar clusters que ocorram para um conjunto particular de dimensões [Parsons et al., 2004].

Dessa forma, há estratégias diferentes que podem ser usadas para dados de alta dimensionalidade. Entre elas, está uma classe de algoritmos voltada para a clusterização em subespaço [Parsons et al., 2004; Woo et al., 2004; Gan et al., 2007], que tem como meta a identificação de clusters cujos dados são próximos para subconjuntos específicos de dimensões do espaço amostral. Portanto, essas técnicas analisam a similaridade entre os dados em conjuntos específicos de dimensões, que muitas vezes se sobrepõem, e tais conjuntos caracterizam o que é chamado de subespaço do espaço de dados [Parsons et al., 2004]. Porém, a investigação de cada uma das possíveis combinações de subespaços tornaria a estratégia inviável computacionalmente e, portanto, há diferentes maneiras apresentadas na literatura para se procurar por subespaços que fazem sentido para um certo conjunto de dados, sem a necessidade de se recorrer à busca exaustiva.

Portanto, os desafios das técnicas de clusterização em subespaço são: (i) identificar os clusters válidos, sendo que a quantidade estimada de configurações possíveis, segundo Xu & Wunsch [2009] e Liu [1968], para se dividir N objetos em K clusters

não-vazios é dada pela Equação 2.1; e (ii) levando-se em consideração a quantidade de subespaços que podem ser gerados, a qual, para um conjunto de D dimensões, pode ser calculada pela Equação 2.2.

$$P(N,K) = \frac{1}{K!}\sum_{m=1}^{K}(-1)^{K-m}C_m^K m^N \tag{2.1}$$

$$S(D) = 2^{D-1} - 1 \tag{2.2}$$

em que $S(D)$ representa todas as formas possíveis de se combinar D dimensões em subespaços, não sendo admitido um subespaço sem dimensões. Assim, no caso de dados de alta dimensionalidade gerados por descritores de imagem como o SIFT, que produz vetores de características com 128 dimensões, deve-se decidir entre $2^{127} - 1$ subespaços possíveis (cálculo da ordem de 10^{38}).

Outro ponto importante no projeto de algoritmos de clusterização, como foi dito anteriormente, é a escolha do critério de análise da similaridade entre dois pontos no espaço de dados. No caso de algoritmos de clusterização hierárquicos, nos quais o espaço de dados é dividido repetidamente ou, por outro lado, pontos individuais vão sendo agrupados para formar os clusters, a aplicação das funções de distância é suficiente. Entretanto, para dados de alta dimensionalidade analisados por algoritmos de clusterização particionais ou baseados em *kernel*, que buscam por regiões densas no espaço de dados, apenas as funções de distância não têm a capacidade de fornecer subsídios para a análise dos dados, sendo necessária a adoção de funções objetivo que, otimizadas, irão guiar a detecção das regiões de maior densidade [Xu & Wunsch, 2009].

Parsons et al. [2004] apresentam um estudo detalhado de várias técnicas de clusterização em subespaço, bem como a divisão dessas técnicas, que inicialmente podem ser consideradas: *top–down*, quando o conjunto completo dos dados é avaliado e as divisões ocorrem de acordo com a avaliação da dissimilaridade entre os pontos, e *bottom–up*, quando cada instância dos dados é observada e os agrupamentos ocorrem segundo critérios de similaridade. Adicionalmente, as técnicas são novamente classificadas conforme o que os autores chamam de medidas de localidade [Parsons et al., 2004].

Segundo Parsons et al. [2004], uma forma de se avaliar a localidade dos dados nos métodos *bottom–up* é através do uso de grades divisórias do espaço de dados que, na prática, se apresentam na construção de histogramas dos dados, nos quais cada intervalo representa a contagem de pontos para uma determinada dimensão. Em seguida, os clusters são descobertos pela avaliação da densidade em conjuntos fixos de intervalos ou dimensões, chamados algoritmos baseados em grade estática, ou agrupa-

mentos variáveis de dimensões, conhecido como algoritmos de grade adaptativa. Para esses algoritmos, de grade estática ou adaptativa, é crítica a definição de um limiar de densidade que determina o agrupamento de regiões densas, por possibilitar a fusão de clusters distintos ou a divisão de um único cluster em dois ou mais, indevidamente.

Para os métodos *top–down*, é inicialmente feita uma divisão do conjunto de dados através de sementes que representam os centros ou centroides de cada cluster. Posteriormente, em cada iteração dos algoritmos, é avaliado o peso associado a cada cluster, seja pela avaliação global da contribuição de cada ponto pertencente ao cluster, ou pela avaliação local da vizinhança dos pontos pertencentes ao cluster [Parsons et al., 2004]. O resultado da avaliação dos pesos faz com que os centroides se desloquem para as regiões mais densas, tornando-se importante a definição do número inicial de clusters e o tamanho dos subespaços [Parsons et al., 2004].

Finalmente, em Parsons et al. [2004], dois algoritmos são comparados: *Merging of Adaptive Finite Intervals* (MAFIA) [Goil et al., 1999] e *Fast and INtelligent Subspace Clustering Algorithm using DImension VoTing* (FINDIT) [Woo et al., 2004]. Os autores do artigo indicam o FINDIT como o de melhor desempenho no quesito escalabilidade, tendo entre suas principais características o uso de um esquema de votação para definir a escolha dos clusters, a aplicação de uma medida de distância orientada por dimensões e a definição explícita do conjunto de dimensões que define cada um dos clusters.

Posteriormente, Patrikainen & Meila [2006] exploram diversas medidas de distância adequadas aos algoritmos de clusterização em subespaço e estabelecem quatro novas medidas para comparação dos clusters gerados por diferentes algoritmos: Erro de Clusterização (*Clustering Error* (CE)), Índice Rand (*Rand Index* (Rand)), Variação da Informação (*Variation of Information* (VI)) e Área Relativa de Não-Interseção (*Relative NonIntersecting Area* (RNIA)).

Mais recentemente, Kriegel et al. [2009] realizam um estudo comparativo de diversos algoritmos de clusterização voltados para dados de alta dimensionalidade. Um dos primeiros pontos importantes discutidos no artigo é a inviabilidade do uso de algoritmos de redução de dimensionalidade para problemas de clusterização, uma vez que a maioria das técnicas de redução de dimensionalidade é global, ou seja, não avaliam o subespaço onde residem os pontos e podem gerar uma perda importante de características locais dos mesmos.

Kriegel et al. [2009] abordam a análise de dados de alta dimensionalidade em duas etapas: descoberta dos subespaços e busca dos clusters. Sendo assim, os algoritmos analisados são classificados segundo as técnicas empregadas para executar cada uma das etapas, o que torna o resultado da classificação dos algoritmos diferente da

proposta por Parsons et al. [2004], que segue a classificação tradicional dos algoritmos de clusterização. O método de classificação tradicional é também usado por Xu & Wunsch [2009] para apresentar diversas técnicas de clusterização mas, principalmente, aquelas baseadas no consagrado método *K–means*.

Neste trabalho, serão descritos três algoritmos de clusterização, sendo que o algoritmo MAFIA será abordado de forma breve, apenas a título de comparação com o FINDIT, que será uma das técnicas usadas na realização dos experimentos, seguindo a indicação feita por Parsons et al. [2004]. Por último, o método de clusterização chamado de *Mean-Shift for Subspace Clustering* (MSSC) [Gan et al., 2007] será detalhado.

O algoritmo MAFIA é baseado no uso de uma grade adaptativa para tentar descobrir os clusters presentes em um determinado conjunto de dados, sendo que a densidade de pontos dentro de cada célula da grade é usada para determinar a localização dos clusters [Goil et al., 1999]. Entretanto, de acordo com Parsons et al. [2004], apesar de o poder de escalabilidade do algoritmo MAFIA ser superior ao do FINDIT, quando analisadas bases de dados da ordem de milhares de pontos, o segundo se destaca quando a ordem de grandeza dos conjuntos atinge milhões de pontos, por usar uma amostragem guiada da base de dados. Além disso, para uma grande quantidade de dimensões, os dois algoritmos apresentam erros na detecção dos clusters e na associação correta das dimensões relevantes aos respectivos clusters. Como este trabalho irá lidar com bases de pontos da ordem de milhões, gerados por descritores de alta dimensionalidade a partir de imagens de bases de cenas urbanas, o algoritmo FINDIT foi a primeira escolha.

Para comparação direta com o algoritmo FINDIT nos experimentos foi escolhido o algoritmo MSSC [Gan et al., 2007]. Trata-se de uma técnica de clusterização baseada em *kernel* que calcula, para cada ponto da base de dados, a probabilidade dele pertencer a um determinado cluster. Além disso, o algoritmo MSSC seleciona, inicialmente, uma amostra aleatória do conjunto de dados para estimar os centroides dos clusters, centroides esses que são refinados iterativamente até que ocorra sua estabilização e o processo de clusterização seja encerrado.

A seguir, os dois algoritmos empregados neste trabalho são detalhados e mais tarde, na seção de experimentos, serão comparados através do uso de uma base sintética que foi gerada de acordo com o protocolo proposto por Aggarwal et al. [1999] e parametrizada segundo os testes realizados por Woo et al. [2004].

2.4.1 FINDIT

A proposta do algoritmo FINDIT é ser rápido e preciso na descoberta de clusters em subespaços e suas dimensões significativas, ou seja, as dimensões que determinam o subespaço no qual aquele cluster existe [Woo et al., 2004]. Além do desempenho no tempo de processamento, o projeto do algoritmo FINDIT para a localização de *outliers* é outro fator importante para sua escolha. O fato dessa técnica de clusterização usar uma amostragem da base de dados possibilita agilidade na análise inicial da informação e a precisão, segundo os autores, fica a cargo do uso de uma função de cálculo da distância entre os pontos orientada a dimensões.

O algoritmo calcula a distância entre dois pontos quaisquer empregando a distância orientada por dimensões — *dimension-oriented distance* (dod) [Woo et al., 2004]. O cálculo de distância proposto pelos autores avalia, basicamente, a distância Manhattan ou norma L_1 nas dimensões significativas dos subespaços com os quais os pontos estão relacionados. Para o resultado final, são levadas em consideração as diferenças inferiores a um limiar ϵ.

Na Equação 2.3, a distância dos pontos $p \rightarrow q$ é avaliada com base nas dimensões significativas do subespaço onde p está, ou seja, as dimensões que definem aquele subespaço. Em seguida, para cada dimensão que também faça sentido no subespaço do ponto q é calculada a diferença entre os dois pontos. Se essa diferença for inferior a ϵ, a dimensão é abatida do total de dimensões significativas do subespaço de p. Portanto, a medida *dod* é uma contagem do número de dimensões para as quais a distância entre dois pontos é inferior a ϵ. Sendo assim, quanto mais próxima de 0, melhor.

$$dod_\epsilon(p \rightarrow q) = |D_p| - |\{d||p(d) - q(d)| \leq \epsilon,\ d \in D_p \cap D_q\}|, \tag{2.3}$$

Ao final, a distância entre dois pontos quaisquer, p e q, é dada por:

$$dod_\epsilon(p, q) = max\{dod_\epsilon(p \rightarrow q), dod_\epsilon(q \rightarrow p)\}$$

Uma vez estabelecida a medida de similaridade entre pontos, a clusterização consome oito etapas de processamento, desde a análise inicial dos dados até a associação final de cada item aos clusters encontrados. Na Figura 2.3, os passos do algoritmo são representados da seguinte forma: o Passo 1, pela Fase de Amostragem; os passos de 3 a 7, pela Fase de Formação de Clusters; e o Passo 8, pela Fase de Associação de Dados. O FINDIT recebe como parâmetros iniciais $C_{minsize}$, que determina a quantidade mínima de pontos que um cluster deve conter, e $D_{mindist}$, a distância máxima entre dois clusters

para que esses sejam concatenados, ou ainda, a quantidade de dimensões para as quais a distância calculada entre dois clusters deve estar abaixo de ϵ.

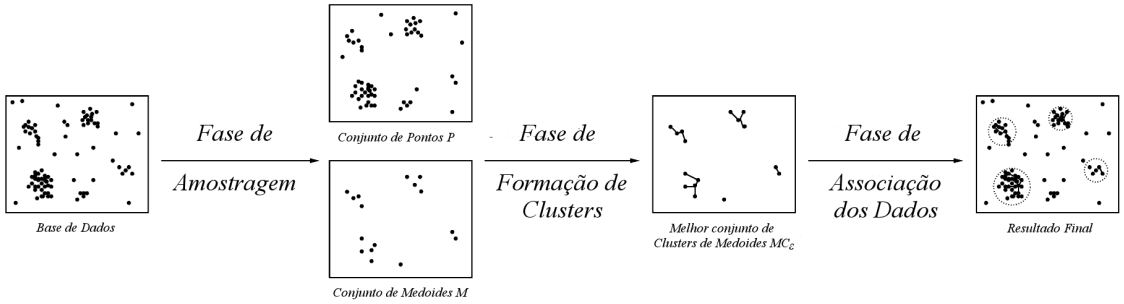


Figura 2.3: Esquema geral de funcionamento do algoritmo FINDIT, reproduzido de Woo et al. [2004]

1. Amostragem

   A base de dados é, inicialmente, amostrada e separada em dois conjuntos: um conjunto maior, de exemplos da base de dados S, e um conjunto menor, de medoides M, que diferem dos centroides por serem pontos reais da base de dados, ao invés de pontos estimados [Xu & Wunsch, 2009]. Os autores garantem a representatividade desses conjuntos, em relação à base de dados completa, através do uso dos Limites de Chernoff, cuja ideia é determinar o número mínimo de itens que devem ser colhidos da base de dados de forma a garantir que todos os clusters, ainda desconhecidos, estejam representados por, pelo menos, ξ itens. O valor de ξ trata-se, então, de um parâmetro implícito do algoritmo, que os autores recomendam que seja ajustado para 30. O valor de $C_{minsize}$ é usado para estimar a quantidade de clusters k e, posteriormente, ambos são empregados no cálculo de ρ, como pode ser visto na Equação 2.4.

$$\textit{Chernoff bounds } (S) = \xi k\rho + k\rho \log\left(\frac{1}{\delta}\right)$$

$$+k\rho\sqrt{\left(\log\left(\frac{1}{\delta}\right)\right)^2 + 2\xi \log\left(\frac{1}{\delta}\right)}, \tag{2.4}$$

$$k = \frac{N}{C_{minsize}}$$

$$\rho \; satisfaz \; C_{minsize} = \frac{N}{k \times \rho} (\rho \geq 1) \therefore$$

$$\rho = \frac{N}{k \times C_{minsize}} = 1,$$

Assumindo que a faixa de valores dos pontos para cada dimensão está normalizada e é representada por r, Woo et al. propõem o uso de 25 diferentes valores de ϵ compreendidos entre $[(1/100)r, (25/100)r]$, que irão gerar configurações distintas de clusters, avaliadas no Passo 7.

2. Escolha das Dimensões Chave

   Após a seleção dos dois subconjuntos, os dados amostrados são utilizados para definir as dimensões chave dos pontos em M, ou seja, aquelas que caracterizam os subespaços, segundo a função de similaridade proposta pelos autores. Para cada item em M, é selecionado um conjunto V de vizinhos mais próximos em S que atuarão como eleitores na determinação das dimensões chave do referido medoide. $|V|$ é outro parâmetro implícito no algoritmo, para o qual os autores recomendam o uso do valor 20. A distância usada para a escolha dos $|V|$ vizinhos em relação a cada elemento de M é calculada pela Equação 2.5.

$$dod_\epsilon = |D| - |\{d||m(d) - p(d)| \leq \epsilon, \; d \in D\}|, \tag{2.5}$$

   em que $p(d)$ é o valor do item p na dimensão d, $|D|$ é o número de dimensões dos dados e ϵ é a maior distância aceita entre um medoide e um elemento em S, numa dimensão d.

   Sendo assim, Woo et al. [2004] provam que para 20 vizinhos v de um ponto m, se 12 ou mais pontos em V estão próximos de m na mesma dimensão d, essa dimensão pode ser considerada uma dimensão chave daquele medoide.

3. Associação de Membros

   Cada elemento $p \in S$ é associado a um medoide $m \in M$, de acordo com a menor distância aferida para cada dimensão chave do medoide, definida no passo anterior. O objetivo é que a distância dod_ϵ de cada medoide para cada elemento em S seja menor que o limiar ϵ em todas as dimensões chave, ou seja, $dod_\epsilon(m \rightarrow p) = 0$. Se um elemento de S atinge essa situação para mais de um medoide, ele será associado àquele que tiver uma maior quantidade de dimensões chave.

4. Clusterização dos Medoides

   De acordo com o parâmetro $D_{mindist}$, medoides ou conjuntos de medoides são concatenados. Apesar do nome, é importante levar em consideração que $D_{mindist}$ define, na verdade, a quantidade máxima de dimensões em que distância entre os medoides analisados pode ser superior à ϵ (Equação 2.6).

$$dod_\epsilon\left(m_i, m_j\right) = max\left\{|D_{m_i}|, |D_{m_j}|\right\} - \left|d\right| |m_i\left(d\right) - m_j\left(d\right)| \leq \epsilon, d \in D_{m_i} \cap D_{m_j}\Big|, \tag{2.6}$$

   Nesse ponto, apesar de ser uma técnica de clusterização de particionamento, o algoritmo FINDIT assume o comportamento de um algoritmo de clusterização hierárquico [Gan et al., 2007].

   A fim de evitar que pares de medoides com no máximo duas dimensões correlacionadas sejam *clusterizados*, quando, na verdade, eles estão distantes um do outro, o valor de dod_ϵ é ajustado para $|D|$, como penalização.

   Para se calcular a distância entre clusters de medoides é empregada a Equação 2.7.

$$dod_\epsilon\left(mc_A, mc_B\right) = \frac{\sum\limits_{m_i \in mc_A, m_j \in mc_B} \left(|m_i||m_j| dod_\epsilon\left(m_i, m_j\right)\right)}{\left(\sum\limits_{m_i \in mc_A} |m_i|\right)\left(\sum\limits_{m_j \in mc_B} |m_j|\right)}, \tag{2.7}$$

   em que $|m_i|$ representa o número de membros do cluster de medoides mc_A e $|m_j|$, o número de membros do cluster de medoides mc_B. A clusterização hierárquica dos clusters de medoides ocorre até que a distância calculada para cada par seja superior a $D_{mindist}$.

5. Ajuste dos Clusters de Medoides

   Após a clusterização dos medoides, as novas dimensões chave dos clusters formados são identificadas, tendo em vista a associação final dos pontos da base de dados completa aos clusters. O cálculo das novas dimensões chave pode ser visto na Equação 2.8.

$$avg_d = \frac{\sum_{m_i} \delta_i |m_i|}{\sum_{m_i} |m_i|}, \tag{2.8}$$

em que m_i representa cada medoide de um cluster de medoides e δ_i assume o valor 1, se d é uma dimensão chave de m_i, ou 0, se d não for uma dimensão chave de m_i. Os autores estabelecem o limiar de $avg_d \geq 95\%$ para que a dimensão d seja considerada dimensão chave do cluster de medoides formado.

Em seguida, dada a possibilidade de alteração das dimensões chave, os clusters de medoides são analisados par a par, conforme os critérios estabelecidos no Passo 4, para uma nova rodada de clusterização e ajuste das dimensões chave, se necessário.

6. Refinamento

   Grupos de medoides com nenhum ou poucos pontos de S associados a eles são eliminados sob a alegação de que seus clusters originais podem ter uma quantidade de membros inferior a $C_{minsize}$.

7. Avaliação da Qualidade dos Clusters

   A qualidade dos clusters formados é avaliada através do critério de *soundness*, ou seja, o somatório do produto da quantidade de dimensões chave pelo número de membros de cada cluster (Equação 2.9).

$$Soundness(MC_\epsilon) = \sum_{mc \in MC_\epsilon} (|mc| \times |KD_{mc}|), \tag{2.9}$$

   em que $|mc|$ é o número de membros de cada cluster de medoides para um determinado valor de ϵ e $|KD_{mc}|$ a quantidade de dimensões chave de cada cluster de medoides.

   O *soundness* irá indicar, então, a qualidade dos agrupamentos produzidos com a aplicação de um determinado valor de ϵ. Esse valor é aumentado progressivamente, o que significa que os elementos de S podem estar, a cada iteração, mais distantes dos medoides, e os passos de 3 a 6 são repetidos. Ao final, é escolhida a formação de clusters com o maior valor para o cálculo do *soundness*.

8. Associação Final dos Dados da Base

   Finalmente, cada elemento da base de dados é associado a um cluster do conjunto de melhor *soundness*, observando cada medoide do conjunto e as dimensões chave.

Uma importante qualidade desse algoritmo é o uso seletivo dos dados, o que o torna adequado para grandes bases de dados. Entretanto, a associação final dos dados aos clusters encontrados é um ponto negativo, pois, dependendo do número de

medoides, esta fase pode demandar mais tempo de processamento que a descoberta dos clusters.

O algoritmo FINDIT foi implementado na linguagem C e comparado com o MSSC, discutido a seguir. Os resultados desta comparação podem ser vistos na Seção 4.1.2.

2.4.2 MSSC

De acordo com Gan et al. [2007], o algoritmo MSSC é uma extensão do algoritmo *Maximum-Entropy Clustering* (MEC), baseado no conceito físico de liberação de energia durante um processo de recozimento simulado. Esse tipo de processo se baseia em aquecer um determinado material a ponto de permitir um certo grau de liberdade aos seus elementos constituintes e, durante um resfriamento controlado cujo desafio é descobrir a taxa adequada de decréscimo da temperatura, permitir que os elementos constituintes do material se reorganizem de forma a produzir um composto mais puro e minimizando a energia liberada. Portanto, o objetivo do algoritmo MEC é minimizar a energia liberada durante o processo de clusterização e isso é assimilado pelo MSSC [Gan et al., 2007] para o processamento em subespaço. No algoritmo MEC, a energia liberada é calculada pela Equação 2.10 [Rose et al., 1990; Gan et al., 2007].

$$F = -\left(\frac{1}{\beta}\right) \sum_{x \in X} \ln \left(\sum_{j=1}^{k} e^{-\beta \|x - z_j\|^2} \right), \tag{2.10}$$

em que x representa cada elemento em $X = \{x_1, x_2, ..., x_n\}$, z_j é o centroide do cluster C_j, k determina a quantidade inicial de centroides a ser escolhida aleatoriamente na base de dados para pesquisa dos clusters e β está relacionado com a quantidade de clusters que serão encontrados.

Como apontado por Gan et al. [2007], β é um dos parâmetros chave na formulação do algoritmo. Se $\beta = 0$, apenas um cluster será encontrado. Por outro lado, quanto maior o valor de β, maior a probabilidade do número clusters encontrados representar a divisão natural dos dados. A análise da quantidade de clusters descobertos pode ser facilitada com o uso do parâmetro k. Se o número de clusters encontrado for inferior a k, um ou mais centroides serão similares entre si e apenas um, dos similares, será usado na associação final dos dados. Caso contrário, o número de clusters pode ser igual ou superior a k, que nesse caso estará limitando a quantidade de clusters descobertos. Sendo assim, uma nova execução do algoritmo, com um valor maior para k, irá solucionar a indecisão.

A partir da Equação 2.10, verifica-se que o algoritmo MSSC é um método de clusterização baseado em kernel e, nesse caso, um kernel Gaussiano, que possui determinadas propriedades que irão garantir o sucesso do método e serão discutidas após a apresentação das principais características do algoritmo. Adicionalmente, a equação original do algoritmo MEC é estendida para a inclusão da ponderação sobre a associação entre os pontos e os clusters (Equação 2.11).

$$F_{\alpha,\beta}(W,Z) = -\left(\frac{1}{\beta}\right)\sum_{i=1}^{n}\ln\left(\sum_{j=1}^{k} e^{-\beta\sum_{h=1}^{d} w_{jh}^{\alpha}(x_{ih}-z_{jh})^2}\right), \tag{2.11}$$

em que W é a matriz de pesos que associa um peso w_{jh} para cada dimensão h do centroide z_{jh}, Z é a matriz de centroides e α é o termo *"fuzzificador"*, ou seja, que vai interferir na probabilidade de cada ponto pertencer a um ou mais clusters.

Portanto, α é outro parâmetro importante no algoritmo, sendo um controlador *fuzzy* dos pesos das dimensões. O α varia entre $(1,\infty)$ e é responsável por destacar as dimensões que formam os subespaços dos clusters durante as iterações do algoritmo, a exemplo do que ocorre com as dimensões chave no FINDIT. O parâmetro α é usado no cálculo da matriz *fuzzy* que indicará a associação final entre os dados da base e os clusters e é indicado por Hathaway & Bezdek [2001] como amplamente aceito como $\alpha = 2$.

O *modus operandi* do algoritmo MSSC pode ser visto na Figura 2.4. Resumidamente, os centroides iniciais são escolhidos da base de dados (porção superior direita da figura) e armazenados na matriz de centroides Z, para a qual é criada a matriz de pesos W. A partir dessas duas matrizes, é criada a matriz *fuzzy* U e, seguindo a sequência numérica, as três matrizes são calculadas iterativamente até que a diferença entre as matrizes de centroides de dois passos adjacentes seja inferior a um limiar ϵ. Finalmente, a matriz Z resultante é usada para associar os elementos da base de dados aos clusters encontrados (porção inferior direita da figura). As etapas do algoritmo MSSC são detalhadas a seguir, sendo k a quantidade esperada de clusters e $|D|$ a quantidade de dimensões dos dados analisados.

1. Inicialização das matrizes de Centroides (Z) e Pesos (W)

   Inicialmente, k pontos da base de dados são escolhidos aleatoriamente e armazenados na matriz de Centroides $Z_{k\times|D|}$. Uma matriz de Pesos $W_{k\times|D|}$ é criada para armazenar o peso de cada dimensão para cada centroide. No início, todas as dimensões têm o mesmo peso e recebem o valor $(1/|D|)$.

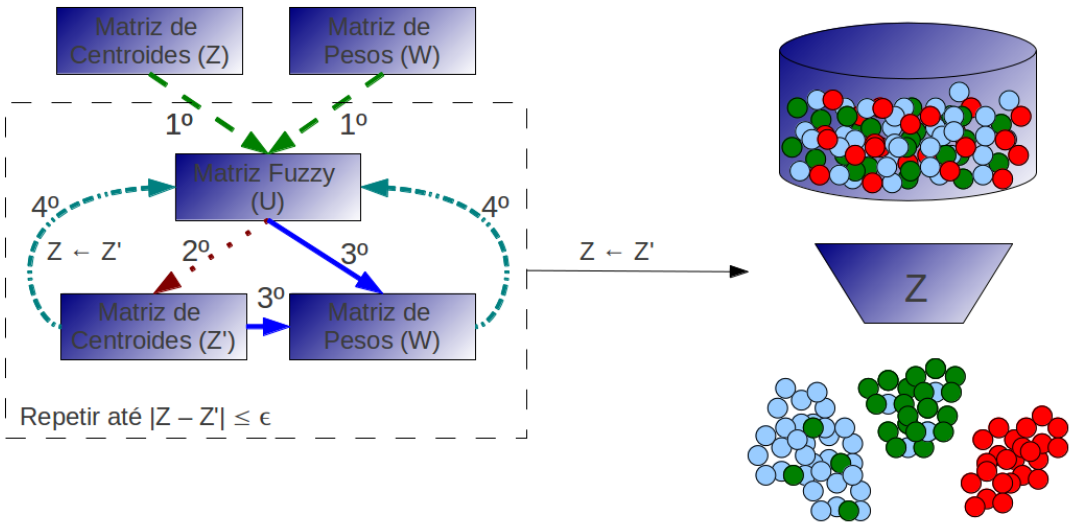


Figura 2.4: Esquema geral de funcionamento do algoritmo MSSC

2. Inicialização da matriz *Fuzzy U*

Uma matriz *Fuzzy* $U_{k\times N}$ é calculada a partir das matrizes Z e W. Sua função é armazenar a probabilidade que cada ponto tem de pertencer a um dos k clusters. Os elementos u_{ji} da matriz U são calculados, segundo Gan et al. [2007], pela Equação 2.12.

$$u_{ji} = \frac{e^{-\beta \sum_{h=1}^{d} (w_{jh})^{\alpha}(x_{ih}-z_{jh})^2}}{\sum_{l=1}^{k} e^{-\beta \sum_{h=1}^{d} (w_{jh})^{\alpha}(x_{ih}-z_{jh})^2}}, 1 \leq j \leq k, 1 \leq i \leq n \tag{2.12}$$

O cálculo da matriz U se baseia no conceito de conjuntos difusos. Esse conceito foi discutido por Zadeh [1965], que abordou os fundamentos das funções que mapeiam elementos para classes segundo um fator de associação entre [0,1], sendo que quanto mais próximo da unidade o fator de associação de um elemento x_i para uma classe C_j estiver, maior a probabilidade desse elemento pertencer a essa

classe. As primeiras aplicações do fator de associação *fuzzy* para reconhecimento de padrões foram sugeridas por Bellman et al. [1966] e em algoritmos de clusterização por Ruspini [1969] e por Dunn [1973]. Assim, mantendo-se a mesma ideia proposta anteriormente, cada elemento u_{ji} representa, na verdade, o valor de $P(x_i \in C_j)$ e é baseado na contribuição da distância do ponto x_i para o centroide z_j em relação ao somatório da distância do mesmo ponto x_i para o centroide de cada um dos clusters C_j. Dessa forma, o valor de u_{ji} é inversamente proporcional à distância do ponto ao cluster e garante- se que $\sum_{j=1}^{k} u_{ji} = 1, 1 \leq i \leq n$, que é a normalização dos fatores de associação, e $\sum_{i=1}^{n} u_{ji} > 0, 1 \leq j \leq k$, que é a impossibilidade da ocorrência de clusters vazios.

3. Atualização das Matrizes

   Após sua criação, as três matrizes são atualizadas iterativamente, na seguinte ordem:

   - A matriz Z é atualizada a partir de U e armazenada em Z', de acordo com Gan et al. [2007]:

   $$z^*_{jh} = \frac{\sum\limits_{i=1}^{n} x_{ih} P(x_i \in C_j)}{\sum\limits_{i=1}^{n} P(x_i \in C_j)} = \frac{\sum\limits_{i=1}^{n} x_{ih} u_{ji}}{\sum\limits_{i=1}^{n} u_{ji}}, 1 \leq j \leq k, 1 \leq h \leq |D| \qquad (2.13)$$

   Indicando, basicamente, que os centroides serão recalculados segundo a densidade dos dados em sua vizinhança, uma vez que estão presentes em u_{ji} os elementos que garantem que os pontos no mesmo subespaço de z_{jh} terão mais peso.

   - A matriz W é recalculada usando-se as matrizes U e Z', seguindo a formulação proposta por Gan et al. [2007]:

   $$w^*_{jh} = \frac{1}{\sum\limits_{l=1}^{|D|} \left[\frac{\sum\limits_{i=1}^{n} u_{ji}(x_{ih} - z_{jh})^2 + \epsilon}{\sum\limits_{i=1}^{n} u_{ji}(x_{il} - z_{jl})^2 + \epsilon} \right]^{\frac{1}{\alpha - 1}}}, 1 \leq j \leq k, 1 \leq h \leq |D| \qquad (2.14)$$

A matriz de pesos W é inicializada com cada dimensão tendo a mesma probabilidade $(1/|D|)$ e, em sua atualização, é observada a densidade dos pontos x_i em torno do centroide z_j em cada dimensão em relação à densidade dos pontos em todas as dimensões para o mesmo centroide, garantindo que

$\sum_{j=1}^{k} w_{jh} = 1, 1 \leq h \leq |D|$. Além disso, w_{jh} é inversamente proporcional à distância dos pontos x_i ao centroide z_j numa dada dimensão h. O parâmetro α aparece como fator de normalização, uma vez que, no cálculo da função objetivo, w_{jh} aparece elevado à potência α. Assumindo que $u_{ji} > 0$, conforme a Equação 2.12, e que, consequentemente, não há clusters vazios, o fator ϵ que evita a ocorrência de divisões por 0 pode ser retirado da formulação.

A cada iteração, é verificado se $|Z - Z'| \leq \epsilon$. Sendo verdadeira a condição, o processo é interrompido e a matriz Z' é inspecionada para a retirada de centroides similares e, após a retirada dos centroides similares, Z recebe os valores de Z'. Caso contrário, Z recebe os valores de Z' e uma nova iteração é iniciada.

O parâmetro ϵ, usado para avaliar a distância entre os centroides, tem como valor sugerido 1×10^{-5} [Gan et al., 2007] e pode ser alterado de acordo com as características de cada base. Por exemplo, se cada centroide for representado apenas por valores inteiros, uma diferença inferior ao valor unitário é suficiente para indicar a semelhança entre dois pontos.

4. Associação final dos Dados da Base

   Por último, cada elemento x_i da base é associado a um dos centroides z_j de Z segundo os valores u_j definidos na matriz U, ou seja, em cada coluna dessa matriz, a linha que contém o maior valor indica o cluster ao qual o elemento deve ser associado. Percebe-se, portanto, que a definição geral do algoritmo MSSC, apesar de utilizar os fatores de associação *fuzzy*, realiza uma associação rígida (*hard assignment*) de membros ao final do processo de clusterização.

As provas de minimização da função objetivo, apresentada na Equação 2.11, através dos cálculos empregados para os valores de u_{ji}, z_{jh}, w_{jh} são discutidas amplamente por Gan et al. [2007].

Além disso, o kernel Gaussiano usado na função objetivo do algoritmo MSSC consegue garantir a convergência dos centroides para as regiões onde há o aumento máximo da densidade dos dados [Comaniciu & Meer, 2002]. De acordo com Cheng [1995], a convergência é garantida, ainda, pelo fato do conjunto de centroides ser distinto do conjunto de dados. Caso o conjunto de centroides pertencesse ao conjunto de dados, a alteração dos centroides a cada iteração poderia modificar a distribuição de densidade dos dados e, consequentemente, exigir garantias extras para a conversão do método.

Adicionalmente à convergência, o êxito do algoritmo MSSC pode ser atribuído ao uso do multiplicador de Lagrange β, que aparece na função objetivo do algoritmo

MSSC. Esse parâmetro foi introduzido, inicialmente, na formulação da solução da clusterização por entropia máxima MEC [Rose et al., 1990], cujo objetivo é associar pontos x e clusters C_j calculando o custo médio para cada arranjo possível de clusters, conforme a Equação 2.15 [Rose et al., 1990].

$$\langle E \rangle = \sum_{x} \sum_{j} P(x \in C_j) E_j(x), \tag{2.15}$$

em que $E_j(x)$ denota o custo de associar o ponto x ao cluster C_j. Sendo assim, o objetivo dos autores é escolher a configuração de clusters que maximiza a entropia do conjunto, de forma que as probabilidades da Equação 2.15 podem ser calculadas por distribuições de Gibbs [Rose et al., 1990]:

$$P(x \in C_j) = \frac{e^{-\beta E_j(x)}}{Z_x}, \tag{2.16}$$

em que a função de partição Z_x é dada por Rose et al. [1990]:

$$Z_x = \sum_{k} e^{-\beta E_k(x)} \tag{2.17}$$

Posto isso, Rose et al. [1990] não só confirmam a análise feita por Gan et al. [2007], de que quanto maior o valor de β maior a probabilidade dos clusters descobertos refletirem uma clusterização natural dos dados, como também associam esse parâmetro à temperatura do processo de recozimento simulado, sendo um inversamente proporcional ao outro, ou seja, manter o valor de β em zero significa manter a temperatura do sistema alta e, portanto, levará à localização de um mínimo global que será o centro da massa dos dados. Por outro lado, o aumento dirigido do valor de β esbarrará em um ponto de transição a partir do qual o único mínimo global será dividido em vários mínimos locais, ou seja, uma temperatura de resfriamento na qual as sementes ou centroides iniciais convergirão para diferentes clusters. Tendo sido a questão da convergência discutida anteriormente, o algoritmo MSSC se mostra indicado para a análise de dados reais [Comaniciu & Meer, 2002]. Além disso, Rose et al. [1990] encaminham a relação entre o valor de β e a associação *fuzzy* entre os pontos da base de dados e os clusters, isto é, para $\beta = 0$, cada ponto está igualmente associado a todos os clusters, associação suave (*soft assignment*), aumentando-se o grau *"fuzzificador"* do sistema; e, na direção do aumento do valor de $\beta \to \infty$, cada ponto terá a tendência de ser associado a um único cluster, ou associação rígida (*hard assignment*), diminuindo-se o grau *"fuzzificador"* do sistema.

A relação do parâmetro β com as configurações possíveis dos clusters para o algoritmo MSSC fica clara através da análise das expressões de Gibbs nas Equações 2.16 e 2.17, nas quais a função de custo $E_j(x)$ pode ser substituída pelo cálculo da distância entre o ponto x e o centroide do cluster C_j. Dessa forma, ele garante que a convergência se dará para os pontos de máxima densidade e, consequentemente, onde o gradiente em torno do centroide será 0, ou seja, um mínimo local. Portanto, a energia liberada ao final do processo será mínima já que os centroides estarão nos pontos de máximo equilíbrio dentro do conjunto de dados.

Basicamente, a maior desvantagem do MSSC é o seu alto custo de execução. Isso acontece porque o algoritmo carrega todos os dados necessários em memória *Random Access Memory* (RAM) e, além disso, os cálculos de atualização das matrizes são muito caros.

A partir do ferramental teórico apresentado até aqui, discute-se, a seguir, alguns dos trabalhos relacionados com a identificação de cenas que servirão de base para as contribuições propostas neste trabalho.

2.5 Identificação de Cenas

O algoritmo clássico para identificação de objetos e cenas é baseado em votação sobre o casamento de descritores locais e pode ser encontrado, por exemplo, na aplicação do descritor SIFT descrita por Lowe [2004]. Inicialmente, calcula-se a distância Euclidiana dos descritores de uma base de imagens para os descritores de um conjunto de imagens de busca, sendo feita a indexação dos descritores da base de dados em relação aos das imagens de busca, segundo suas distâncias. Já nesse momento, os autor destaca a possibilidade da correspondência incorreta entre as características locais devido à ambiguidade dos descritores ou a descritores provenientes do plano de fundo da imagem, que podem ser caracterizados como descritores de baixa qualidade ou de baixo poder de discriminação. A transformada de Hough é então aplicada para refinar os casamentos entre os descritores, procurando por grupos de três associações entre descritores de um mesmo objeto e a mesma imagem da base de dados.

Outras restrições geométricas, como o algoritmo RANSAC [Fischler & Bolles, 1981], podem ser impostas para a redução de falsos positivos nesse esquema de votação. Valle et al. [2009] e Picard et al. [2009] avaliam a confiabilidade dessas técnicas para cenas *street–view*, sendo realizada a mesma avaliação inicial de distância entre descritores e aplicado o algoritmo de indexação Multicurves [Valle et al., 2008], como requisitos prévios da fase de avaliação da consistência geométrica. Os autores con-

cluem que os casamentos de descritores não retornam resultados confiáveis quando há muitos falsos positivos entre os descritores locais. A presença de muitas características locais com baixa capacidade de discriminação é apontada como uma das principais dificuldades para uma identificação de alta qualidade de cenas urbanas. Realizando experimentos numa base de dados *street-view* da cidade de Paris, é mostrado que essas características locais com baixa capacidade de discriminação são prejudiciais para a acurácia da identificação de cenas urbanas. Além disso, Valle et al. [2009] e Picard et al. [2009] indicam que a maioria das características não discriminantes são geradas por oclusões (especialmente por objetos altamente texturizados, como árvores), sombras e fachadas de edificações com elementos muito texturizados.

Turcot & Lowe [2009] abordam a tarefa de identificação de objetos em bases de dados contendo uma grande quantidade de elementos. O método BoW [Sivic & Zisserman, 2003; Csurka et al., 2004] (vide Seção 5.1.1) é a solução usual para essa questão. A proposta dos autores é a escolha de um conjunto específico de características locais ao invés do espaço completo de descritores da imagem, que contém dados espúrios ou discriminantes o suficiente. É mostrado no trabalho que 4% das características originais por imagem são suficientes para garantir uma performance para o casamento de imagens com a mesma taxa de acertos obtida para a base completa. Segundo os autores, o uso do método BoW reduz o uso de memória em, no máximo, uma ordem de magnitude, dependendo do número de características dentro de cada palavra visual. Sendo assim, o termo característica útil é definido como aquela que é robusta o suficiente para ser casada como uma característica correspondente no mesmo objeto, estável o suficiente para existir em vários pontos de vista e distinta o suficiente para que as características correspondentes sejam associadas à mesma palavra visual.

O trabalho anterior apresenta avanços nos resultados obtidos inicialmente por Philbin et al. [2007], que compara diferentes métodos para criação do vocabulário visual através do uso de dois algoritmos de clusterização: *Approximate K-means* (AKM) e *Hierarchical K-means* (HKM). Ambos são versões otimizadas do tradicional algoritmo de clusterização K–means, o qual não é escalável para bases de dados da ordem de 5 000 ou 10 000 imagens [Philbin et al., 2007]. Em seguida, restrições geométricas são impostas no intuito de prever as transformações afins existentes entre as imagens de busca e as imagens da base de dados. Dessa forma, os autores conseguem um sistema de identificação de cenas escalável, cujos resultados são avaliados para a base de dados de Edifícios de Oxford (*Oxford Buildings*), também proposta por Philbin et al. [2007].

Diferente dos métodos que apresentam o uso de descritores locais extraídos das imagens no domínio espacial, na proposta feita por Oliva & Torralba [2001], as imagens são analisadas no domínio da frequência para extrair o sentido básico ou *Gist* de ima-

gens da natureza. As imagens são analisadas, portanto, de forma global e as imagens de consulta reconhecidas de acordo com suas assinaturas no domínio da frequência.

Para permitir a recuperação de informação visual em tempo real, Torralba et al. [2008] propõem o uso de técnicas de aprendizagem de máquina para converter os descritores Gist [Oliva & Torralba, 2001] para um código binário compacto. Através dos experimentos, mostra-se a eficácia da representação compacta de descritores para a recuperação de dados em tempo real e em grandes bases de dados.

Liu et al. [2009] propõem a descrição semântica de cenas usando uma versão modificada do descritor SIFT *flow* [Liu et al., 2008], chamada *coarse-to-fine SIFT flow*, combinada com o descritor Gist [Oliva & Torralba, 2001]. O objetivo é transferir as anotações presentes na base de dados para a imagem de busca usando as melhores correspondências entre pontos e entre as imagens. Para gerenciar as múltiplas sugestões de anotação, é usado um modelo *Markov Random Field* (MRF) para concatenar indicações redundantes em anotações plausíveis.

Enquanto a identificação de cenas preconiza a geração de assinaturas das imagens para que sejam reconhecidas as cenas similares, a classificação usa as mesmas assinaturas para associar imagens a classes de objetos ou situações, que será objeto de estudo da Parte II desta tese.

2.6 Considerações

Neste capítulo foram discutidos os principais conceitos relativos aos descritores de imagem, sejam eles globais como os propostos inicialmente por Bimbo [1999] ou locais como o SIFT [Lowe, 2004], um dos mais citados na literatura. Além disso, abordou-se a clusterização ou, mais precisamente, a clusterização em subespaço, com o intuito de usá-la na proposta de filtragem de descritores para melhoria da acurácia na identificação de cenas.

O objetivo dos descritores é gerar uma assinatura para a imagem que permita identificá-la de forma inequívoca e que, dessa forma, contribua para sua comparação entre outras imagens no intuito de se identificar imagens similares.

Entretanto, as assinaturas das imagens não estão imunes a ruídos, que ocorrem devido à presença de descritores com baixo poder discriminativo. Por isso, planeja-se usar os algoritmos de clusterização em subespaço para a filtragem de tais descritores. A eliminação desses descritores permite, em tese, o aumento da acurácia na identificação de cenas. Dessa forma, diversos autores propõem soluções para a filtragem dos

descritores, em uma fase de pré-processamento, com o objetivo de aumentar a acurácia de seus algoritmos na identificação de cenas.

A seguir, apresentamos uma nova proposta de filtragem desses descritores de baixo poder discriminativo, bem como modificamos o algoritmo MSSC de forma que o mesmo seja adequado à tarefa de filtragem de descritores.

Capítulo 3

Contribuições Propostas para a Identificação de Cenas

Para a separação dos descritores discriminativos daqueles considerados de baixo poder discriminativo, é oferecida uma metodologia de identificação de cenas por meio da filtragem não-supervisionada desses descritores, ou seja, empregando-se a clusterização em subespaço. Além disso, após a análise dos dois algoritmos de clusterização em subespaço estudados em profundidade, os algoritmos FINDIT e MSSC, é proposta uma modificação deste último, que se mostrou mais adequado à tarefa de filtragem dos descritores que o algoritmo FINDIT, mas exigindo, no mínimo, o dobro do tempo de processamento, levando-se em consideração a mesma base de dados.

3.1 Identificação de Cenas

O problema da identificação de cenas é formalmente definido por uma cena $\hat{Q}$, alvo da identificação, uma base de imagens $I = \{\hat{I}_1, \hat{I}_2, ..., \hat{I}_N\}$ e uma função de similaridade, $\delta(\hat{A}, \hat{B})$, a partir dos quais deseja-se obter, inicialmente, uma lista ordenada das imagens em I, tal que

$$Lista_{\hat{I},\hat{Q}} = \{\hat{I}_{j_1}, \hat{I}_{j_2}, ..., \hat{I}_{j_k}\}, \delta(\hat{I}_{j_1}, \hat{Q}) \leq \delta(\hat{I}_{j_2}, \hat{Q}) \leq ... \leq \delta(\hat{I}_{j_k}, \hat{Q}), 1 \leq j_k \leq N \quad (3.1)$$

Em seguida, com base na relação das imagens de I que correspondem às respostas válidas para uma consulta Q, ou seja, a verdade-terrestre de Q, encontra-se o conjunto resposta R.

$$\begin{aligned} &R = Lista_{\hat{I},\hat{Q}} \cap V_{\hat{Q}}, V_{\hat{Q}} = \{\hat{I}_{j_1}, \hat{I}_{j_2}, ..., \hat{I}_{j_l}\}, 1 \leq j_l \leq N, \\ &\text{Sujeito a:} \\ &R = \left\{ \hat{I}_j | \hat{I}_j \in V_{\hat{Q}}, \sum_{\hat{I}_j \in V_{\hat{Q}}} \delta(\hat{I}_j, \hat{Q}) \leq \sum_{\hat{I}_k \notin V_{\hat{Q}}} \delta(\hat{I}_k, \hat{Q}) \right\} \end{aligned} \tag{3.2}$$

A Equação 3.2 estabelece que a eficácia da solução do problema da identificação de cenas está sujeita ao retorno das imagens pertencentes à verdade-terrestre de Q, $V_{\hat{Q}}$, de forma que as distâncias das imagens desse subconjunto R de I para $\hat{Q}$ sejam inferiores às distâncias das imagens em I que não pertencem a $V_{\hat{Q}}$.

Apesar dessa ser a solução ideal para o problema, o que se consegue, na prática, são imagens de R cuja distância para a cena que se deseja reconhecer é superior às distâncias obtidas em relação a imagens que não pertencem à verdade-terrestre da cena em questão. Além disso, é possível afirmar que os principais responsáveis por esse efeito são os descritores usados no processo de obtenção das distâncias.

Na identificação de cenas, o grande volume de descritores gerados para as imagens, ao invés de vantajoso, se mostra prejudicial, uma vez que para poucos descritores discriminantes, aqueles que produzem casamentos verdadeiros entre imagens, há uma grande quantidade de descritores de baixa qualidade, ou que produzem casamentos equivocados entre imagens, oriundos de regiões de oclusão, sombras e regiões altamente texturizadas [Valle et al., 2009; Picard et al., 2009] ou, ainda, de descrições ambíguas [Lowe, 2004]. Esse problema é abordado por Turcot & Lowe [2009] que, através de um treinamento supervisionado, reduzem os descritores em 96% para cada imagem, ao mesmo tempo que mantêm a taxa de acerto na identificação de cenas.

3.2 Identificação de Cenas pela Filtragem de Descritores Locais

Separar os descritores discriminantes dos não-discriminantes ou, simplesmente, filtrar os descritores é um passo crítico para que a identificação de cenas seja bem sucedido. A técnica proposta envolve o uso da clusterização em subespaço para separar descritores em clusters segundo sua similaridade. Posteriormente, é feita uma busca por esses clusters de maneira a identificar qual é aquele que contém apenas descritores discri-

minantes. Naturalmente, o objetivo é melhorar a eficácia da recuperação de imagens semelhantes em uma base, a partir do uso de uma imagem de busca.

O protocolo consiste em usar um conjunto de imagens de busca para definir os ranques médios de recuperação das imagens corretas em uma base de imagens, estabelecendo-se, assim, os valores de referência. Para isso, são usados algoritmos de casamento de imagens que fazem parte do estado da arte em recuperação de informação para a identificação de cenas. No caso deste trabalho, é empregado o algoritmo proposto por Valle et al. [2009] e, também, por Picard et al. [2009], no qual os descritores de uma imagem da base de dados próximos aos descritores de uma imagem de busca são convertidos em votos para a associação das duas imagens. Posteriormente, algumas correspondências são eliminadas através da aplicação de restrições geométricas como o RANSAC.

Em seguida, os descritores das imagens da base de dados são processados por um algoritmo de clusterização em subespaço e cada cluster gerado é usado para a identificação de cenas. Os resultados produzidos são comparados com os valores de referência para a identificação do cluster sem a presença de descritores não discriminantes. Dessa forma, identifica-se um subconjunto do espaço de descritores em que a identificação de cenas atinge índices superiores aos valores de referência usando uma quantidade bem menor de descritores. O funcionamento da técnica proposta é apresentado na Figura 3.1 que, na área pontilhada superior, apresenta o protocolo tradicional para identificação de cenas. Na área pontilhada inferior, o novo protocolo é apresentado, com a fase de clusterização da base de cenas urbanas ocorrendo antes do processamento pelo algoritmo de casamento de imagens. Na sequência, cada cluster é confrontado com a base de imagens por meio do algoritmo de casamento de imagens e o ranque obtido para cada cluster é comparado com o ranque médio do protocolo tradicional, permitindo a indicação do cluster que contém os dados que melhoram o desempenho da identificação de cenas.

Permanece, ainda, o desafio de se identificar, automaticamente, o cluster que contém a melhor filtragem de descritores empregando, para isso, medidas de qualidade sobre os clusters produzidos.

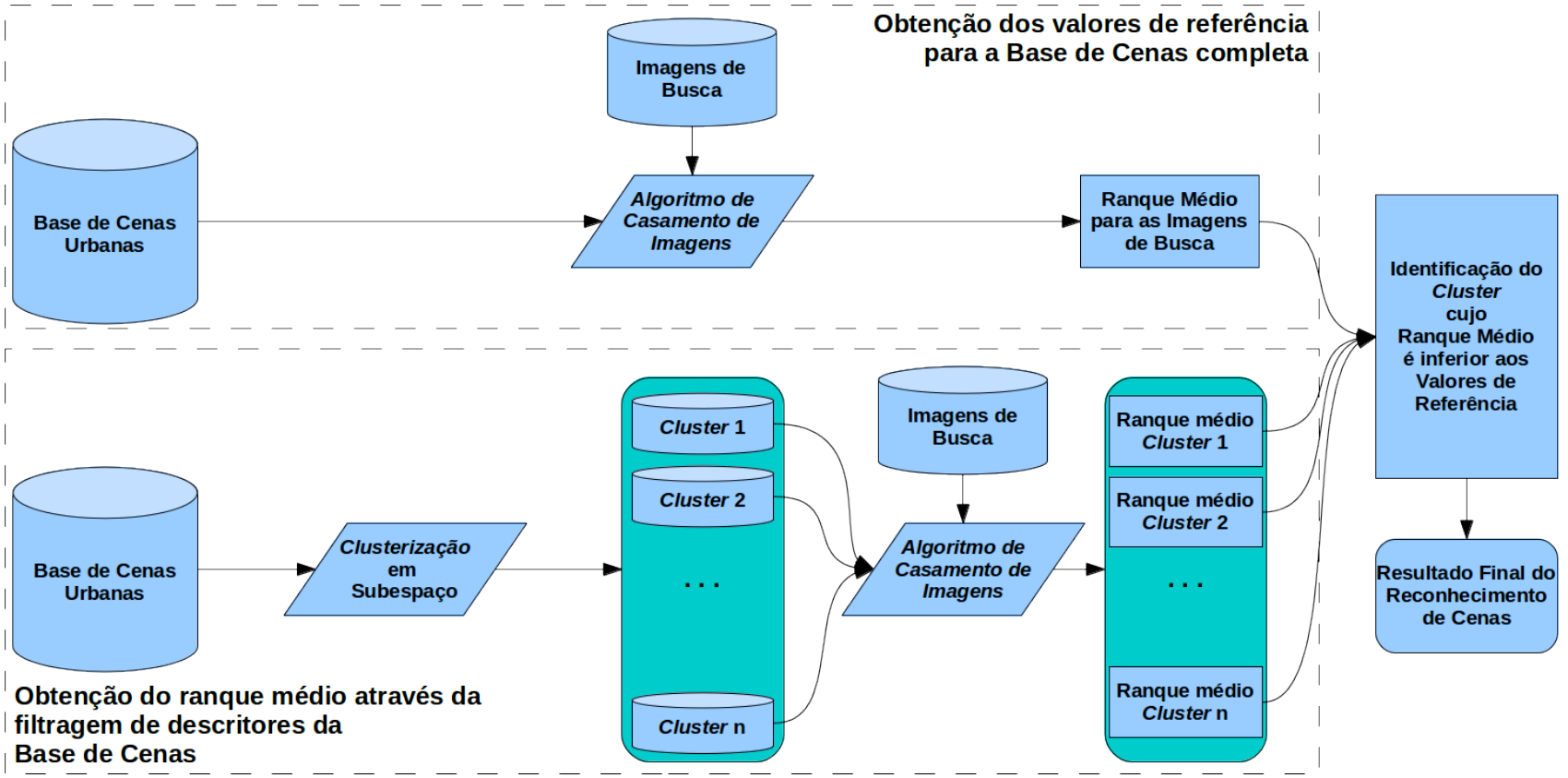


Figura 3.1: Esquema geral de funcionamento da técnica de aplicação da clusterização em subespaço para melhorar a precisão dos algoritmos de casamento de imagens.

3.3 Algoritmo MSSC usando Amostragem — E–MSSC (Enhanced MSSC)

Na comparação dos algoritmos MSSC e FINDIT através de bases sintéticas, na Seção 4.1.2, percebeu-se um desempenho próximo dos dois algoritmos, mas com uma leve vantagem para o algoritmo MSSC, cujo erro total de classificação ficou em torno de 0,001%. Entretanto, o algoritmo MSSC é cerca de duas vezes mais lento que o algoritmo FINDIT, sendo este um problema indesejável para grandes bases de dados. Optou-se, portanto, pelo desenvolvimento de uma variação do algoritmo MSSC que prezasse pela otimização do tempo de execução.

Para isso, a aprendizagem baseada em amostras aparece como alternativa: a estratégia é aplicar o algoritmo de clusterização MSSC em uma pequena amostra aleatória de uma base de dados e, posteriormente, estender o resultado obtido com a amostra para a base de dados completa. Assim, uma versão aprimorada do MSSC é proposta, contendo dois novos passos: uma fase de pré-processamento, para obter a amostra dos dados em uma quantidade cuidadosamente calculada; e uma fase de pós-processamento, que amplia os resultados da clusterização para a base de dados original. Esta versão aprimorada do algoritmo de clusterização recebeu o nome *Enhanced Mean-Shift for Subspace Clustering* (E–MSSC) e pode ser observada no Algoritmo 1.

A principal vantagem de se utilizar a amostragem dos dados se reflete na complexidade do algoritmo MSSC original, dada por $O(|DB| \times |D| \times k)$, em que $|DB|$ é a quantidade de pontos da base de dados DB, $|D|$ é a quantidade de dimensões e k o número inicial de clusters. Deve-se notar que esta complexidade é atingida durante o cálculo da matriz U, no qual é avaliada a probabilidade de cada um dos $|DB|$ pontos da base pertencer a um cluster C_k, representado pelo centroide z_k, avaliando-se a distância do ponto a cada um dos k centroides em todas as $|D|$ dimensões. A amostragem não altera a complexidade do algoritmo, mas tem um impacto importante no tempo de execução do mesmo, reduzindo a quantidade de dados analisada. Sendo assim, o uso de uma amostra equivalente a 17,28%$|DB|$ representa uma redução teórica de 82,72% no tempo de execução. Entretanto, isso resultou, na prática, em uma redução de 99% no tempo de execução para o melhor caso.

3.3.1 Fase de Amostragem

Para se amostrar a base de dados antes do processo de clusterização, é importante determinar o tamanho adequado da amostra, objetivando reduzir o uso de recursos computacionais enquanto se preserva a representatividade da população. Para a clus-

Entrada:
DB : base de dados de descritores,
$|D|$: quantidade de dimensões dos descritores,
k : número estimado de clusters,
c : tamanho mínimo de cada cluster,
s : quantidade mínima de representantes de cada cluster na amostragem,
α : *fuzzyficador*,
β : regulador do número final de clusters.
Saída:
C : matriz $|DB| \times 2$ que associa cada ponto em DB a um cluster,
Z : matriz $\lceil k \rceil \times |D|$ de centroides,
W : matriz $\lceil k \rceil \times |D|$ de pesos,
U : matriz $|DB| \times \lceil k \rceil$ *fuzzy*.

```
1  //Cálculo do volume de dados a ser amostrado.
2  size ← ChernoffBounds(DB, c, s);
3  DB' ← Amostra(DB, size);
4  //Aplicação do algoritmo MSSC sobre a base de dados amostrada.
5  (C, Z, W, U) ← MSSC(DB', k, α, β);
6  //Reconstrução das matrizes U e W para a base de dados completa;
7  U' ← Recalcula(DB, Z, W);
8  W' ← Recalcula(DB, Z, U');
9  //Descoberta das dimensões chave a partir da matriz W';
10 kd[|Z|][|D|] ← ExtraiDimensõesChave(W');
11 //Associação dos pontos da base de dados aos centroides, comparando
12 //as distâncias apenas nas dimensões chave.
13 para todo p ∈ DB faça
14 |   para todo z ∈ Z faça
15 |   |   se Mínimo(dist(p, z, kd)) então
16 |   |   |   C ← Associa(p, z)
17 |   |   fim
18 |   fim
19 fim
20 retorna C,Z,W',U'
```

Algoritmo 1: E–MSSC

terização baseada em densidade, a representatividade da amostra é obtida quando se mantém uma alta probabilidade de pelo menos uma pequena parcela de cada cluster ser amostrada [Guha et al., 1998]. Dessa forma, clusters contendo uma quantidade de pontos muito pequena são geralmente desprezados e considerados como *outliers*.

A presença de cada exemplar dos dados em um cluster particular pode ser modelada como um teste de Bernoulli (ou seja, o ponto pertence ao cluster ou não), e a quantidade total de pontos do cluster a serem amostrados pode ser modelada após a soma desse testes de Bernoulli. Esse raciocínio permite uma estimativa do tamanho da amostra através dos *Chernoff bounds*, aplicados de forma a garantir que uma fração grande o suficiente dos clusters seja amostrada com alta probabilidade. Esse método, proposto por Guha et al. [1998] para o algoritmo de clusterização Cure, foi, posteriormente, empregado pelo algoritmo de clusterização FINDIT [Woo et al., 2004]. Até onde se sabe, este é o primeiro trabalho em que os *Chernoff bounds* serão aplicados com o algoritmo MSSC.

Para que os *Chernoff bounds* determinem o tamanho da amostra, alguns parâmetros são necessários: o número estimado de clusters k na base de dados; um fator ρ indicando quão grande deve ser o tamanho médio dos clusters (em número de pontos) em relação ao tamanho do menor cluster a ser considerado (isto é, se o tamanho médio dos clusters é de 1000 pontos e o tamanho do menor cluster a ser considerado é de 100 pontos, $\rho = 10$); a quantidade mínima de representantes ξ que se quer para cada cluster — este valor deverá ser ajustado para determinar a fração esperada de pontos no caso de grandes clusters; e a probabilidade máxima δ da amostragem não cumprir tais limites. Então, o tamanho da amostra é dado pela Equação 3.3 [Guha et al., 1998].

$$s_{min} = k\rho \left[\xi - ln\delta + \sqrt{(ln^2\delta - 2\xi ln\delta)} \right] \tag{3.3}$$

O tamanho da amostra cresce linearmente com o número de clusters esperados k e na proporção inversa ρ do menor cluster. Assim, se a base de dados cresce porque todos os clusters se tornam proporcionalmente maiores, o tamanho da amostra não muda. Por exemplo, se $k = 100$, $\rho = 5$, $\xi = 16$ e $\delta = 0{,}001$, uma amostragem de 20 000 pontos será suficiente, independentemente do tamanho da base de dados.

Para o êxito do E–MSSC, assume-se, também, que o processo de amostragem não interfere na densidade da distribuição dos dados, podendo, no máximo, atenuar a densidade de pontos em algumas regiões. Dessa forma, o funcionamento geral do algoritmo MSSC não é prejudicado, conforme discutido na Seção 2.4.2.

Uma vez que a representatividade da amostra está garantida, o algoritmo MSSC é aplicado nessa amostra e os clusters encontrados são estendidos para o restante da

base de dados. Sabendo-se que os clusters fazem sentido apenas nos subespaços e que o algoritmo MSSC atribui graus *fuzzy* de associação tanto para os pontos como para as dimensões, essa extensão dos clusters para todos os pontos da base deve ser conduzida com extremo cuidado, como explicado a seguir.

3.3.2 Extensão da Clusterização

Inicialmente, deve-se lembrar que o algoritmo MSSC tem como saída três matrizes: os centroides dos clusters (Z), os pesos de pertinência para as dimensões dos clusters (W), e o fator de associação *fuzzy* de cada ponto aos clusters (U). As duas primeiras permitem a avaliação da distância ponderada entre os centroides e os pontos da base de dados e a última permite realizar as associações de forma determinística, atribuindo cada ponto ao cluster cujo centroide é o mais próximo.

No aprimoramento que está sendo proposto para o algoritmo, a matriz de centroides Z e a matriz de pesos W, ambas originadas do processo de amostragem dos dados, são usadas para construir uma nova matriz U para a base de dados completa. No intuito de acelerar a associação final dos dados, as dimensões chave de cada centroide são usadas.

Essas dimensões chave são descobertas na matriz de pesos W após observar que os valores dos pesos para as dimensões chave são muito superiores aos pesos para as outras dimensões. Adicionalmente, os pesos são normalizados pelo operador log, e os que estão acima da média indicam as dimensões chave.

Fazer com que o subespaço do cluster seja manipulado explicitamente é uma adaptação crucial em relação ao algoritmo inicial, de forma a garantir, após a fase de extensão dos clusters, a compatibilidade entre os clusters estendidos e os gerados originalmente pelo MSSC sobre a base de dados amostrada.

3.4 Considerações

Após a formalização da recuperação de informação visual na identificação de cenas foram discutidas as principais contribuições deste trabalho nessa área.

Para a identificação de cenas, foi introduzida a etapa de filtragem dos descritores pelo uso da clusterização em subespaço, seguida do protocolo convencional de identificação de cenas [Valle et al., 2009; Picard et al., 2009] para cada cluster e seleção manual do melhor resultado, quando comparado com o valor de referência obtido para a identificação pré-filtragem, ou seja, empregando-se toda a base de descritores.

Com o intuito de reduzir o tempo necessário para a fase de clusterização e, consequentemente, da filtragem dos descritores, o algoritmo E–MSSC foi proposto. O novo algoritmo faz uso de uma amostragem guiada da base de descritores, garantindo um desempenho na descoberta de clusters equivalente ao algoritmo MSSC [Gan et al., 2007], com um tempo de processamento expressivamente menor, conforme será mostrado na Seção 4.4.

Capítulo 4

Experimentos

Neste capítulo, é apresentado o protocolo experimental adotado para a identificação de cenas através da filtragem de descritores por clusterização em subespaço, em bases de dados de cenas urbanas. Na sequência, são relacionadas as bases de dados reais empregadas nos testes e, em seguida, os resultados obtidos são detalhados e discutidos.

4.1 Protocolo Experimental

Dada a natureza dos problemas abordados neste trabalho, foi necessária a definição de dois protocolos experimentais, destinados à identificação de cenas e à classificação de imagens. Neste capítulo será abordado o primeiro problema, enquanto o segundo será visto na Capítulo 7.

4.1.1 Identificação de Cenas por Filtragem de Descritores

Durante a revisão do estado da arte, verificou-se que o emprego de algoritmos de casamento de imagens já consagrados na literatura não poupou Valle et al. [2009] e Picard et al. [2009] da obtenção de baixos índices de identificação de cenas. A análise feita nesses trabalhos relata a ocorrência de descritores de baixo poder discriminativo, gerados por regiões altamente texturizadas como árvores e sombras nas imagens da base, como causa do insucesso das identificações.

A hipótese de ocorrência desses descritores é validada através do uso de algoritmos de clusterização em subespaço, com o objetivo de tentar identificar os pontos que apontam a similaridade entre imagens que, na verdade, não são similares e, consequentemente, excluir tais pontos do processo de busca para melhorar o ranque médio

de acerto do algoritmo. Os primeiros resultados da aplicação dessa técnica podem ser encontrados no estudo feito por de M. Coelho et al. [2011].

Os experimentos foram baseados no algoritmo de identificação de cenas empregado por Valle et al. [2009] e Picard et al. [2009], que tem como objetivo o casamento de uma imagem capturada por um telefone celular com as imagens de uma base de dados de cenas urbanas. Resumidamente, o algoritmo usa vetores de características extraídos pelo algoritmo SIFT [Lowe, 2004] e se baseia num esquema simples de votação:

1. Inicialmente, o algoritmo procura pelos vetores de características das imagens da base mais próximos de cada vetor de características de cada uma das imagens de consulta, por meio do algoritmo de indexação *Multicurves* [Valle et al., 2008] e aplicando a distância Euclidiana.

2. Cada vetor de características de uma imagem da base de dados próximo a um dos vetores da imagem de consulta conta como um voto para essa imagem da base de dados.

3. As imagens da base são, então, ranqueadas de acordo com os votos que receberam em ordem decrescente; quanto mais votos receber, mais similar ela é à imagem de consulta e, consequentemente, mais próxima do topo do ranque. A imagem vencedora da votação é rotulada com o número um e as demais vão recebendo rótulos crescentes. Dessa forma, quanto menor for o número que rotula uma imagem, maior a similaridade entre essa imagem e a imagem de consulta. O ranque ideal para imagem correta é representado, portanto, pelo número um. Nas tabelas que apresentam os resultados dos experimentos, esta etapa é indicada pelo nome Votação Bruta.

4. O passo anterior é refinado pelo algoritmo RANSAC, que verifica a consistência geométrica entre os pontos selecionados. Assim, são descartadas respostas com alta incidência de pontos similares, mas que não apresentam a mesma semelhança geométrica com a imagem de busca.

5. Um novo ranque é criado para cada imagem de busca, baseado nos votos obtidos por cada imagem após a aplicação da consistência geométrica. Esta etapa é referenciada pelo nome RANSAC nas tabelas que apresentam os resultados dos experimentos.

O algoritmo descrito é aplicado sobre a base completa de descritores e os descritores das imagens de consulta, para obtenção dos valores de referência e, em seguida, a

base de descritores é submetida a um algoritmo de clusterização em subespaço. Cada cluster de descritores encontrado é então submetido ao algoritmo de identificação de cenas juntamente com os descritores das imagens de busca, sendo comparados os resultados da identificação de cenas para cada um dos clusters com os valores de referência.

Apesar de os clusters detectados terem sido avaliados por várias métricas estatísticas em busca de alguma característica que permitisse prever aquele que pudesse melhorar os resultados do casamento entre imagem de busca e imagens da base, nenhuma ocorrência estatística relevante foi encontrada. Dessa forma, os clusters passaram a ser investigados individualmente, com o algoritmo de identificação de cenas, até que se detectasse aquele que oferecesse o melhor desempenho em relação à base original. Finalmente, os resultados obtidos para os clusters são comparados com os valores de referência para ser identificado o melhor cluster, ou seja, aquele que tem menor quantidade de descritores de baixa qualidade.

Antes da aplicação dos algoritmos de clusterização na filtragem dos descritores, é necessária a verificação do desempenho dos algoritmos FINDIT e MSSC em uma base de dados sintética para que os resultados obtidos nas bases de dados reais sejam validados.

4.1.2 Base Sintética para comparação entre os algoritmos FINDIT e MSSC

Com o objetivo de avaliar os dois algoritmos apresentados anteriormente, uma base de dados sintética foi gerada, empregando os critérios apresentados no estudo de Aggarwal et al. [1999] e os parâmetros definidos por Woo et al. [2004]. As características da base de dados criada são (Figura 4.1a):

a) ela contém 100 000 pontos com 20 dimensões e distribuídos em cinco clusters;

b) não há *outliers* nos dados;

c) o tamanho mínimo definido para cada cluster é de 5 000 pontos e o número de pontos associados a cada cluster segue uma distribuição exponencial, com média 1;

d) o número médio de dimensões correlacionadas é sete (seguindo uma distribuição de Poisson); e

e) a variação definida para a distribuição dos pontos nos clusters foi [2,4], seguindo uma distribuição Normal.

A partir das características da base de dados sintética foram estabelecidos os parâmetros dos algoritmos FINDIT e MSSC usados em sua clusterização.

Para o algoritmo FINDIT, os parâmetros usados foram $C_{minsize} = 5\,000$, $D_{mindist} = 0$, $\xi = 30$ e a quantidade de votantes igual a 20. Os clusters encontrados podem ser vistos na Figura 4.1b e a matriz de confusão na Figura 4.1d.

Nas Figuras 4.1b e 4.1d, é possível perceber que o algoritmo FINDIT classifica alguns pontos como *outliers* e pode ser visto um erro de classificação de 1,852% para o Cluster 1, 0,005% para o Cluster 2 e 0,001% para o Cluster 4.

O algoritmo MSSC, por sua vez, não detecta *outliers* e sua performance pode ser vista na Figura 4.1c, que apresenta os clusters encontrados, e sua Matriz de Confusão, na Figura 4.1e. Os parâmetros usados para a descoberta dos clusters foram $\alpha = 4{,}1$, $\beta = 43{,}9364$ e $k = 10$.

O erro de classificação do método MSSC, 0,001% nos Clusters 1 e 4, é muito menor que o experimentado para a descoberta do Cluster 1 no algoritmo FINDIT e idêntico ao verificado para o Cluster 4, também obtido pelo algoritmo FINDIT. Entretanto, o tempo de execução do algoritmo MSSC se degrada rapidamente com o aumento do volume de dados processado. Portanto, o algoritmo MSSC é mais preciso, mas seu uso se torna crítico no caso de um volume de dados da ordem de milhões de pontos. Por fim, é interessante notar a correta identificação das dimensões dos clusters pelos dois algoritmos testados.

Na próxima seção, serão apresentados os detalhes das bases de dados reais usadas na aplicação de identificação de cenas com filtragem de descritores.

4.2 Bases de Dados Utilizadas nos Testes

Nos testes das aplicações são empregadas bases de cenas urbanas da literatura e, também, bases de dados coletadas durante o desenvolvimento deste trabalho (ver Apêndice A), para permitir a validação das metodologias propostas.

4.2.1 Paris

A base de Paris foi coletada no âmbito do Projeto iTowns[1], para o qual um furgão percorreu as ruas da capital francesa capturando imagens com um conjunto de câmeras. As imagens de fachadas de duas ruas dessa base foram usadas para testar técnicas consagradas na literatura sobre identificação de cenas a partir de imagens de busca

[1] http://www.itowns.fr/

Base Sintética

Cluster	Membros	Dimensões
1	7505	3,7,8,9,19
2	8029	6,7,8,9,11,16,19
3	25311	2,6,7,8,12,16,19
4	11423	1,2,9,12,16,17,19
5	47732	1,2,8,9,19

(a)

FINDIT

Cluster	Membros	Dimensões
1	7366	3,7,8,9,19
2	7986	6,7,8,9,11,16,19
3	25311	2,6,7,8,12,16,19
4	11411	1,2,9,12,16,17,19
5	47864	1,2,8,9,19
Outliers	62	-

(b)

MSSC

Cluster	Membros	Dimensões
1	47733	1,2,8,9,19
2	25311	2,6,7,8,12,16,19
3	11423	1,2,9,12,16,17,19
4	8029	6,7,8,9,11,16,19
5	7504	3,7,8,9,19
Outliers	-	-

(c)

Clusters encontrados pelo FINDIT

	1	2	3	4	5	*Outliers*
1	7366	0	0	0	132	7
2	0	7986	0	0	0	43
3	0	0	25311	0	0	0
4	0	0	0	11411	0	12
5	0	0	0	0	47732	0

(d)

Clusters encontrados pelo MSSC

	5	4	2	3	1	*Outliers*
1	7502	0	0	0	3	-
2	0	8029	0	0	0	-
3	0	0	25311	0	0	-
4	0	0	0	11423	0	-
5	2	0	0	0	47730	-

(e)

Figura 4.1: Resultados dos experimentos com a base sintética (a). Podem ser vistas as matrizes de Clusters e de Confusão para o FINDIT ((b) e (d)) e para o MSSC ((c) e (e)). Ambos detectaram corretamente as dimensões dos subespaços. Enquanto o FINDIT detectou *outliers* inexistentes, o MSSC alcançou o menor erro geral de classificação.

formadas por fotografias tiradas com o uso de um dispositivo móvel [Valle et al., 2009; Picard et al., 2009].

No caso deste trabalho, apenas a base de dados de pior desempenho observado em Valle et al. [2009] e Picard et al. [2009] foi avaliada. Sua escolha se deu justamente pela oportunidade de se testar o método de filtragem de descritores por meio dos algoritmos de clusterização. Essa base de dados é composta de 3 476 087 descritores SIFT [Lowe,

2004] obtidos de 300 imagens, dentro das quais deveriam ser identificadas 10 imagens de consulta representadas por 101 480 descritores SIFT [Lowe, 2004].

4.2.2 Ouro Preto

Esta foi a primeira base das filmagens das cidades históricas a ser separada em imagens e anotada para a tarefa de identificação de uma imagem de pesquisa dentro do banco de dados. Os detalhes da aquisição dos dados dessa base estão no Apêndice A. Essa base de dados faz parte do acervo produzido pelo Projeto Cidade Virtual, financiado pelo CNPq e gerenciado pelo Núcleo de Processamento Digital de Imagens (NPDI).

Para os experimentos de identificação de cenas, foram separadas e anotadas 618 imagens que geraram 1 839 545 descritores SIFT [Lowe, 2004] e construída a verdade-terrestre para 38 imagens de consulta, para as quais foram obtidos 747 250 descritores SIFT [Lowe, 2004]. Essas imagens de consulta foram capturadas a partir de um telefone celular[2].

4.2.3 Edifícios de Oxford (*Oxford Buildindgs*)

A base *Oxford Buildings*[3] foi usada por Turcot & Lowe [2009] para validação da proposta dos autores de selecionar descritores úteis e identificar cenas empregando uma quantidade reduzida de vetores de características em relação ao volume total de descritores disponíveis.

Na constituição da base de dados *Oxford Buildings* são empregadas 5 064 imagens de edifícios de Oxford e 104 789 imagens tiradas do site Flickr, que inserem elementos para confundir a identificação de cenas. São gerados, portanto, 170 762 421 descritores que serão usados para a identificação de 55 imagens de busca, divididas em 11 grupos e representadas por 151 309 descritores. O volume ocupado pelos descritores é, respectivamente, 71 GB e 62 MB.

Os autores destacam que foi possível melhorar a capacidade de casamento das imagens de busca com as da base usando apenas 4% da base original. A base de dados *Oxford Buildindgs* foi escolhida por ter sido empregada inicialmente no trabalho de Philbin et al. [2007] para a identificação de cenas e, por Turcot & Lowe [2009], com uma filtragem de descritores precedendo a identificação de cenas. Além disso, trata-se uma base de dados de cenas urbanas, sendo possível validar a técnica proposta de incremento na identificação de cenas pela filtragem dos descritores através da clusterização em subespaço.

[2]Cooperação com a Escola de Belas Artes da UFMG, na pessoa do Prof. Dr. Alexandre Leão
[3]http://www.robots.ox.ac.uk/ vgg/data/oxbuildings/

A seguir, serão mostrados os resultados obtidos com a aplicação da filtragem de descritores na identificação de cenas para as bases de dados descritas nesta seção.

4.3 Filtragem prévia de descritores para a Identificação de Cenas

Sendo um dos objetivos deste trabalho, a investigação dos benefícios que a técnica de clusterização em subespaço pode trazer para as tarefas de recuperação de informação visual, nesta seção será avaliada sua atuação na filtragem dos descritores empregados em algoritmos de casamento de imagens ou identificação de cenas.

Partimos do princípio que os algoritmos de clusterização estudados anteriormente são úteis na melhoria da performance das técnicas de identificação de cenas, retirando os descritores com baixo poder discriminativo do processo. Os resultados de referência (*baseline*) que serão usados para comparação apresentam as imagens de consulta para uma determinada base de dados, seguidas de valores que indicam a localização da primeira imagem correta na lista de imagens similares, obtidas a partir do uso de todos os descritores disponíveis. É bom lembrar que, no caso dos experimentos realizados, o ranque está associado à similaridade entre a imagem de consulta e suas imagens próximas; sendo assim, quanto menor o valor da imagem no ranque, melhor a classificação realizada pelo algoritmo, se ela for uma resposta correta para a imagem de consulta correspondente. Esses resultados são obtidos por meio do protocolo de identificação de cenas descrito em 4.1.1.

4.3.1 Identificação de Cenas na Base de Dados de Paris

Para a base de dados de Paris, os valores de referência estão na Tabela 4.1. Cada uma das linhas da tabela apresenta o ranque obtido, para a imagem destacada na primeira coluna, em relação à identificação da cena pela Votação Bruta e pelo refinamento desta usando o algoritmo RANSAC. É importante destacar que para a imagem de consulta 9, da Tabela 4.1, não houve, após o refinamento pelo RANSAC, nenhuma imagem da base de dados considerada similar. Nesse caso, a referida imagem de consulta foi desconsiderada do cômputo do ranque médio para a coluna RANSAC na tabela citada. Os valores foram obtidos a partir dos descritores e do código empregado por Valle et al. [2009] e Picard et al. [2009]. Na execução dos experimentos com a filtragem de descritores, foram empregados inicialmente os algoritmos FINDIT e MSSC. Dada a natureza da base de dados, não existia nenhuma informação que possibilitasse a definição dos

Tabela 4.1: Resultados obtidos com a base de dados Original e usados como referência.

Imagens de Busca	Base Original	
	Votação Bruta	RANSAC
1	4	10
2	32	15
3	54	16
4	50	50
5	3	21
6	163	31
7	250	138
8	173	1
9	169	-
10	172	73
Ranque Médio	106,9	39,4
Ganho	-	-

parâmetros livres desses algoritmos. Por isso, foram feitos alguns testes de ajuste de parâmetros onde o melhor resultado foi empregado na filtragem dos descritores.

Os parâmetros livres do algoritmo MSSC são α, que controla o grau de imprecisão da associação entre cada ponto do espaço de dados e um cluster; β, que controla a quantidade de clusters; e k, que determina a quantidade inicial de clusters que devem ser procurados. A estratégia para atribuição dos valores desses parâmetros se baseou, principalmente, no cálculo oferecido por Gan et al. [2007] para estimar o valor de β a partir de α. Como essa fórmula de cálculo analisa a entropia da base de dados, os valores variam conforme a natureza da mesma. Sendo assim, foram avaliados valores de α que limitassem β a um número de, no máximo, três dígitos, partindo de $\alpha = 1{,}1$ (valor mínimo para o mesmo — vide Seção 2.4.2), visto que o mesmo é usado na função objetivo para calcular as matrizes do algoritmo MSSC como expoente, o que resultaria num encarecimento exacerbado da etapa de atualização das matrizes, caso o β ultrapassasse essa ordem de grandeza. Já para a definição do valor de k, o algoritmo MSSC foi executado usando os conjuntos α e β calculados anteriormente até que fosse obtido uma quantidade final de clusters inferior ao valor proposto para k, partindo de $k = 20$, assumindo que havia sido alcançada uma limitação da quantidade de clusters relacionada à natureza dos descritores.

O algoritmo MSSC é usado com os parâmetros $\alpha = 3{,}1$, $\beta = 188{,}2861$ e $k = 25$, com os quais foram encontrados 24 clusters na base de dados, com uma média de 144 836 membros cada um. Após a aplicação do algoritmo de classificação em cada

um dos clusters, os melhores resultados são observados para o cluster composto de 37 129 pontos, portanto, 1,06% da quantidade total de pontos da base de dados original (Tabela 4.2).

Para o ajuste do algoritmo FINDIT, são dois os parâmetros livres: $C_{minsize}$ e $D_{mindist}$. Neste caso, o parâmetro crítico é o $D_{mindist}$, já que ele tem influência direta na descoberta dos subespaços da base de descritores. Sendo assim, seguindo os passos de Woo et al. [2004], foi usado como valor para $D_{mindist}$ a parcela de 10% do total de dimensões dos dados, o que resultou em $D_{mindist} = 13$. Além desse valor, foram testados, ainda, $D_{mindist} = 0$ e $D_{mindist} = 26$. Determinados os valores de $D_{mindist}$, o parâmetro $C_{minsize}$ foi ajustado para 1 000 e 5 000. Realizados os experimentos com as combinações desses valores, foi escolhido o par de melhor resultado, conforme os dois casos citados a seguir.

Usando $C_{minsize} = 5\,000$ e $D_{mindist} = 26$, que determinam que cada cluster tenha, no mínimo, 5 000 pontos e até 26 dimensões possam ser ignoradas no momento de avaliar a distância entre os pontos, foram encontrados três clusters e 14,48% de *outliers*. Entretanto, os resultados de busca não foram melhores que os de referência para nenhum dos clusters, o que pode ser justificado pelo valor de $D_{mindist}$, ou seja, se é permitido que 26 dimensões sejam ignoradas no cálculo das distâncias, o risco de ocorrer uma mistura dos descritores discriminantes com os de baixa qualidade é elevado.

Já na aplicação dos parâmetros $C_{minsize} = 1\,000$ e $D_{mindist} = 13$, foram encontrados 256 clusters e 52,71% de *outliers*, sendo que, após os clusters serem submetidos ao algoritmo de casamento de imagens, conseguiu-se resultados melhores que os de referência. O melhor deles foi para o cluster contendo 6 214 pontos ou 0,18% da base de dados original (Tabela 4.2).

Na Tabela 4.2, houve uma melhora significativa do ranque médio usando os algoritmos MSSC e FINDIT para clusterizar a base de dados original. Observando-se os resultados, verifica-se que é possível encontrar ao menos um cluster capaz de melhorar o ranque médio, tanto naqueles descobertos pelo método MSSC quanto nos encontrados pelo método FINDIT. Entretanto, o ranque médio obtido para a identificação de cenas permaneceu superior a 10,0, comprovando que a tarefa de separação dos pontos que causam os erros de casamento não é, de fato, trivial, como já apontavam os trabalhos de Lowe [2004], Valle et al. [2009] e Picard et al. [2009].

Em outras palavras, parece que os pontos de interesse localizados em árvores e sombras realmente confundem a tarefa de comparação das imagens, mas eles não são os únicos. Para investigar esta hipótese, os vetores de características das imagens de busca foram *clusterizados* com o algoritmo MSSC. Um dos clusters encontrados,

Tabela 4.2: Melhoria da identificação de cenas obtida através do uso dos algoritmos MSSC e FINDIT sobre a base de dados e do algoritmo MSSC sobre as imagens de busca. Os valores positivos indicam a melhoria da identificação da cena, comparada com os valores de referência na Tabela 4.1, em direção ao ranque ideal, ou seja, 1. Os valores negativos indicam o distanciamento para o ranque ideal, medido a partir dos valores de referência. Em #Melhoras, verifica-se o número de cenas para as quais a identificação através dos clusters melhorou. As porcentagens indicam a relação entre o tamanho do cluster de melhor resultado e o tamanho total da base de descritores.

Imagens de Busca	MSSC sobre as imagens da base $\alpha = 3{,}1$ e $\beta = 188{,}2861$ 1,06% dos dados		FINDIT sobre as imagens da base $C_{minsize} = 1\,000$ e $D_{mindist} = 13$ 0,18% dos dados		MSSC sobre as imagens de busca - - 46,72% dos dados	
	V. Bruta	RANSAC	V. Bruta	RANSAC	V. Bruta	RANSAC
1	-12	-13	-44	-11	-	-3
2	-11	4	3	10	6	-1
3	36	8	40	11	8	-41
4	-40	33	-3	-4	-1	37
5	-1	13	-21	10	2	12
6	128	10	162	27	19	20
7	-2	-138	120	75	22	89
8	-48	-90	149	-11	8	-2
9	147	14	128	16	21	-
10	84	35	148	51	-2	-73
#Melhoras	4	7	7	7	7	4
R. Médio	78,9	25,7	38,8	21,3	98,7	21,4
Ganho	26,19%	34,93%	63,70%	40,00%	7,67%	36,79%

contendo 46,72% dos pontos da base de imagens de busca, foi usado no processo de busca das imagens e retornou resultados inferiores aos valores de referência (Tabela 4.2). Entretanto, nas imagens de busca não há árvores nem sombras, e, portanto, comprova-se a existência de outros pontos que causam os erros de classificação.

Adicionalmente, no desenvolvimento deste trabalho, foram vistas algumas pesquisas que empregam a redução de dimensionalidade como etapa de pré-processamento para realizar a busca das imagens. Entretanto, optou-se por realizar a filtragem dos descritores sem interferir na quantidade original de dimensões.

A seguir, com o intuito de validar os resultados obtidos com a metodologia proposta na base de cenas urbanas de Paris, aplica-se a mesma metodologia de filtragem

dos descritores através de algoritmos de clusterização em subespaço para melhorar a identificação de cenas na base de dados de Ouro Preto.

4.3.2 Identificação de Cenas na Base de Dados de Ouro Preto

Descreve-se, agora, todo o processo, feito anteriormente, aplicado na base de dados de Ouro Preto.

Nas Figuras 4.2 e 4.3, podem ser vistos exemplos das imagens da base de dados e das imagens de busca, respectivamente. As primeiras quatro imagens da parte superior da Figura 4.2 estão dispostas na sequência em que foram capturadas, ocorrendo, portanto, alguma redundância. Nas demais linhas, não há adjacência temporal entre os quadros exibidos. Na Figura 4.3 estão representadas as imagens de consulta, que poderiam ser divididas em quatro grupos, conforme as fachadas que identificam: CM = Igreja de Nossa Senhora do Carmo, SF = Igreja de São Francisco de Assis, IN = Museu da Inconfidência e TR = Praça Tiradentes. A divisão citada será usada apenas para ilustrar as imagens de consulta, não sendo levada em consideração na produção dos resultados, ou seja, o ranque médio continua sendo calculado para cada imagem de consulta e não por grupo.

Figura 4.2: Exemplos das imagens coletadas no centro histórico de Ouro Preto.

A escolha dos parâmetros livres dos algoritmos MSSC e FINDIT seguiu os mesmos critérios apresentados para a base de dados de Paris e estão indicados na Tabela 4.3.

Figura 4.3: Exemplos das imagens de busca que representam quatro pontos turísticos da cidade de Ouro Preto.

Na Tabela 4.3, aparece o resultado do casamento das imagens para as 38 imagens de busca. A exemplo do que ocorreu com uma das imagens de consulta da base de dados de Paris, quatro imagens de consulta da base de dados de Ouro Preto não apresentam imagens similares na base de dados após refinamento pelo RANSAC. Ou seja, as imagens têm correspondências retornadas na base de dados para a Votação Bruta, mas quando é empregado o RANSAC para verificação da consistência geométrica dos pontos, nenhuma das prováveis imagens similares indicadas pela Votação Bruta atende aos requisitos.

Após o uso do algoritmo de clusterização em subespaço FINDIT e a tentativa de casamento das imagens com cada um dos 150 clusters encontrados, a melhor performance apurada ocorre para o cluster contendo 0,15% dos descritores originais (Tabela 4.3).

Finalmente, nesse segundo conjunto de experimentos, são achadas algumas pistas sobre os clusters que podem melhorar o valor do ranque médio de classificação. Os

Tabela 4.3: Comparação da melhoria do ranque médio para a base de Ouro Preto usando os clusters encontrados pelos algoritmos MSSC e FINDIT.

38 Imagens de Busca na Base de Ouro Preto			
		Votação Bruta	RANSAC
Valores de Referência	Ranque Médio	141,68	37,76
MSSC $\alpha = 3{,}1$ $\beta = 250$ 28,53% dos dados	Ranque Médio	108,97	27,68
	Ganho	23,09%	26,72%
FINDIT $C_{minsize} = 1\,000$ $D_{mindist} = 13$ 0,15% dos dados	Ranque Médio	41,26	25,95
	Ganho	70,88%	31,29%

clusters encontrados pelo método FINDIT apresentam um grande número de dimensões chave e, aproximadamente, 3 000 elementos. Relembrando, as dimensões chave são aquelas que caracterizam a relação entre cada ponto do espaço de descritores e o cluster ao qual está associado, que é um conceito introduzido pelo algoritmo FINDIT [Woo et al., 2004].

Novamente, a base de dados original é *clusterizada*, mas desta vez pelo algoritmo MSSC, com o objetivo de se comparar os resultados com aqueles obtidos pelos clusters gerados pelo algoritmo FINDIT. Sabendo-se que, neste trabalho, foi feita a incorporação do conceito das dimensões chave no algoritmo MSSC, deve ser observado que o cluster que apresenta o melhor ranque médio contém um baixo número de dimensões chave, em torno de duas ou três. Entretanto, a quantidade de elementos continua por volta de 3 000. É alcançada uma melhoria de 26,72% com o cluster contendo 524 911 pontos ou 28,53% dos descritores originais (Tabela 4.3).

Na Tabela 4.3, em que aparecem os ranques médios para o casamento das imagens após a clusterização, somente uma imagem de consulta não obteve imagem da base similar a partir do cluster descoberto pelo algoritmo FINDIT. Para o algoritmo MSSC, a mesma imagem teve sua similar retornada na terceira posição.

Percebe-se, portanto, que o método de filtragem de descritores apresentado melhora a tarefa de identificação de cenas, identificando imagens individuais de forma mais precisa e usando apenas uma pequena amostra da base de dados original (Figura 4.4). Pretende-se, agora, incorporar o algoritmo de clusterização E–MSSC na filtra-

Figura 4.4: Melhoria no casamento de imagens através do uso da clusterização em subespaço: a linha superior mostra a imagem de busca (primeira imagem à esquerda) e sua similar (oitava imagem) que é a 209ª na lista de imagens similares da base de dados original. Após a aplicação do método proposto, nas oito imagens da linha inferior, a imagem de busca (primeira imagem à esquerda) é identificada como sendo a 3ª (terceira imagem da segunda linha) na lista de similaridade retornada pelo cluster usado.

gem de descritores e validar seus resultados com aqueles obtidos previamente para os algoritmos FINDIT e MSSC.

4.4 Investigação do uso do algoritmo MSSC com Amostragem de Dados

De acordo com o que foi proposto na Seção 3.3, deseja-se, a partir de agora, verificar o desempenho do algoritmo E–MSSC proposto e, posteriormente, aplicá-lo à metodologia de filtragem de descritores.

Para comparação dos algoritmos E–MSSC e MSSC foi gerado um conjunto de bases sintéticas com as seguintes características:

a) elas contêm 100 000 pontos com [32,64,128] dimensões e espalhados entre 5 e 10 clusters;

b) são gerados $[0\%, 25\%, 50\%]$ de *outliers* nos dados;

c) os pontos de cada cluster são distribuídos dentro das seguintes faixas de desvio padrão: $\{[2,4], [3,6], [4,8]\}$; e

d) o número médio de dimensões correlacionadas está entre 1/3, 2/3 e 3/3 do total de dimensões dos pontos da base.

Basicamente, cada um dos parâmetros, começando pelo número de dimensões, é fixado e o restante deles é alterado, permitindo a geração de 162 bases sintéticas.

A comparação entre os algoritmos MSSC original e E–MSSC pode ser vista na Figura 4.5, onde o número de dimensões é mantido fixo enquanto são variadas as dimensões chave, permitindo a comparação dos resultados da clusterização para uma amostra das bases de dados sintéticas geradas. A similaridade é quantificada pela métrica F1–Score que leva em consideração tanto falsos positivos como falsos negativos na compilação dos resultados e foi usada para realizar uma comparação semelhante por Woo et al. [2004].

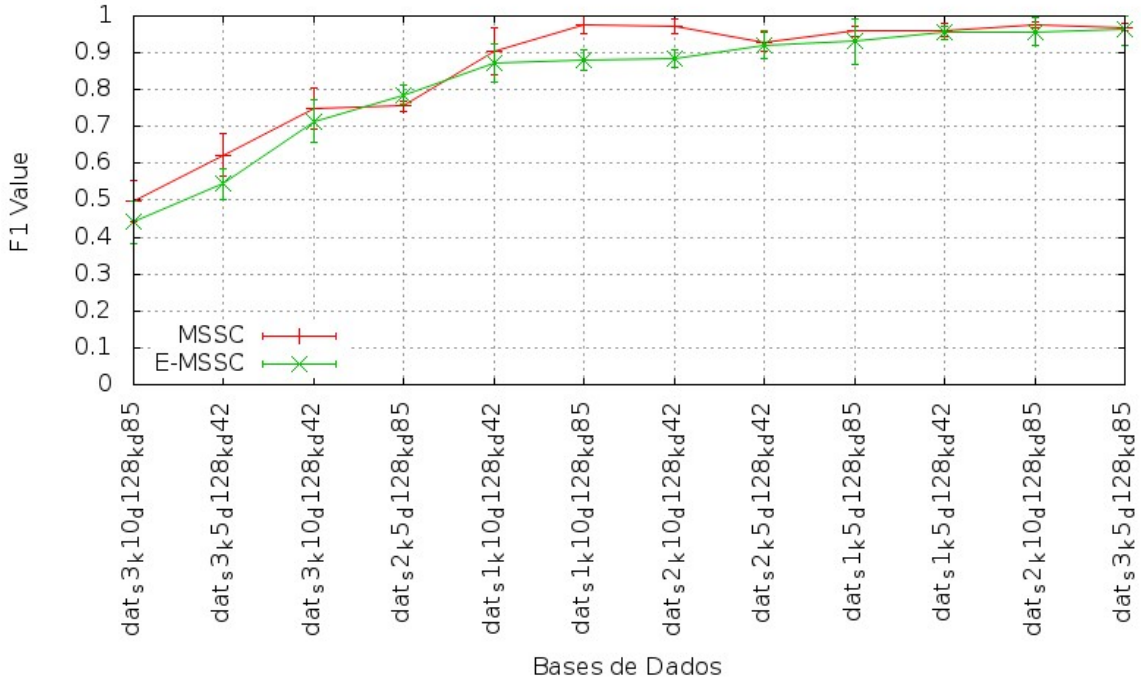


Figura 4.5: Comparação dos algoritmos de clusterização MSSC e E-MSSC usando bases sintéticas, sendo o número de dimensões fixado em $d = 128$ e o de dimensões chave $kd = \{\frac{1}{3}d, \frac{2}{3}d\}$. As barras representam os intervalos para uma confiança de 95%.

Comparado com o algoritmo MSSC original, o algoritmo E–MSSC teve uma performance na descoberta dos clusters de $96{,}41\% \pm 3{,}10\%$, o que compensa a redução do tempo de execução verificada neste último. Ou seja, a grande vantagem do algoritmo E–MSSC é reduzir o tempo de descoberta dos clusters enquanto mantém a qualidade dos mesmos, conforme a comparação feita com o uso das bases sintéticas.

O algoritmo E–MSSC foi usado para avaliar a melhoria da performance na recuperação de imagens para as bases de Paris e Ouro Preto e os resultados podem ser acompanhados pelas Tabelas 4.4 e 4.5. Mais uma vez, os parâmetros livres do algo-

Tabela 4.4: Comparação da melhoria do ranque médio para a base de Paris usando os clusters encontrados pelo algoritmo E–MSSC.

10 Imagens de Busca na Base de Paris			
		Votação Bruta	RANSAC
Valores de Referência	Ranque Médio	106,90	39,40
E–MSSC $\alpha = 3{,}1$ $\beta = 187{,}108$ 18,82% dos dados	Ranque Médio	115,30	**16,00**
	Ganho	-7,86%	**59,39%**
MSSC $\alpha = 3{,}1$ $\beta = 188{,}2861$ 1,06% dos dados	Ranque Médio	78,90	25,70
	Ganho	26,19%	34,93%
FINDIT $C_{minsize} = 1\,000$ $D_{mindist} = 13$ 0,18% dos dados	Ranque Médio	38,80	21,30
	Ganho	63,70%	40,00%

ritmos E–MSSC foram determinados pelos mesmos critérios citados para o algoritmo MSSC.

Apesar de a filtragem pelo algoritmo E–MSSC ter alcançado uma melhor performance na identificação de cenas, no melhor caso e em ambas as bases, nos resultados obtidos não há significância estatística para se afirmar que ele é melhor que os outros dois algoritmos, FINDIT e MSSC. Os resultados de uma comparação direta do algoritmo E–MSSC com o algoritmo FINDIT, que também usa amostragem de dados no processo de clusterização, podem ser vistos na Tabela 4.6

A Tabela 4.7 mostra que, apesar de o ranque médio (24,52 para as duas bases) obtido pelo algoritmo E–MSSC ser ligeiramente superior ao obtido pelo algoritmo FINDIT (21,18 para a base de Paris e 22,37 para a base de Ouro Preto), o ganho em tempo ao se usar a primeira opção compensa a perda em acurácia. Sendo assim, o algoritmo E–MSSC é 62× e 34× mais rápido que os algoritmos MSSC e FINDIT, respectivamente, para a base de imagens de Ouro Preto. Entretanto, para a base de imagens de Paris, que contém um volume três vezes maior de descritores que a anterior, o algoritmo E–MSSC passa a ser 210× mais rápido que o algoritmo FINDIT e 541× mais rápido que o algoritmo MSSC. Em relação ao algoritmo MSSC, o algoritmo E–MSSC é mais rápido por usar uma estratégia de amostragem de dados controlada e, posteri-

Tabela 4.5: Comparação da melhoria do ranque médio para a base de Ouro Preto usando os clusters encontrados pelo algoritmo E–MSSC.

38 Imagens de Busca na Base de Ouro Preto			
		Votação Bruta	RANSAC
Valores de Referência	Ranque Médio	141,68	37,76
E–MSSC $\alpha = 2{,}1$ $\beta = 1{,}68018$ 4,88% dos dados	Ranque Médio	84,16	**21**,**47**
	Ganho	39,19%	**43**,**14**%
MSSC $\alpha = 3{,}1$ $\beta = 250$ 28,53% dos dados	Ranque Médio	108,97	27,68
	Ganho	23,09%	26,72%
FINDIT $C_{minsize} = 1\,000$ $D_{mindist} = 13$ 0,15% dos dados	Ranque Médio	41,26	25,95
	Ganho	70,88%	31,29%

Tabela 4.6: Ranques médios obtidos através da filtragem dos descritores pelos algoritmos E–MSSC e FINDIT, nas bases de Paris e Ouro Preto.

Algoritmo	Paris	Ouro Preto
E–MSSC	24,52	24,52
FINDIT	21,18	22,37

ormente, estender os resultados da clusterização da amostra para a base completa. Já para o algoritmo FINDIT, que usa uma amostragem de dados semelhante, o algoritmo E–MSSC é mais eficiente, sendo capaz de empregar um centroide por cluster, enquanto que o algoritmo FINDIT precisa definir diversos medoides por cluster, tornando mais cara a associação final dos pontos da base de dados aos clusters.

Outro ponto interessante é o fato de que apenas uma parte dos clusters encontrados melhora a identificação das cenas, enquanto que a grande maioria produz resultados até piores para a mesma tarefa, conforme pode ser visto na Figura 4.6, relativa à base de dados de Ouro Preto. Verifica-se, na figura, que do total de 21 clusters encontrados, 28,57% deles, representados por barras verdes, melhoram o ranque médio da identificação das imagens, enquanto que os outros 71,53%, representados pelas barras vermelhas, pioram o ranque médio da identificação de cenas em índices superiores a 10%.

Tabela 4.7: Comparação do tempo de clusterização dos três algoritmos abordados para as bases de Paris e Ouro Preto.

	Base		
Algoritmo	Paris	Ouro Preto	Tempo(s)
MSSC	703 471	44 369	
FINDIT	274 070	24 622	
E–MSSC	1 300	714	

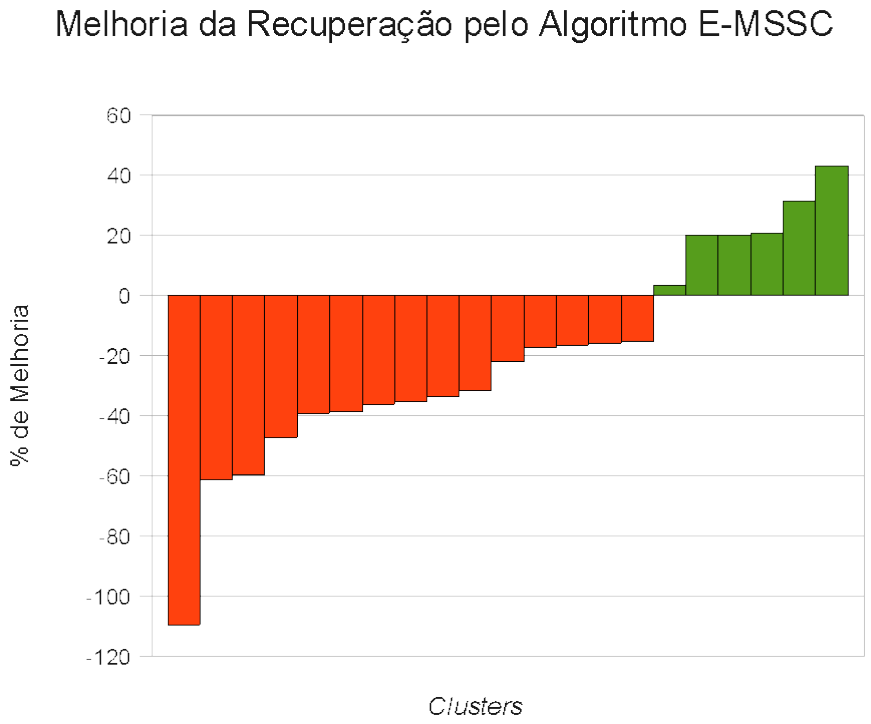


Figura 4.6: Melhoria relativa do ranque médio conseguida pela avaliação de cada cluster resultante da clusterização da base de dados de Ouro Preto pelo algoritmo E–MSSC. Valores positivos indicam melhoria (redução do ranque médio), enquanto que valores negativos indicam o oposto.

Novamente, pode-se perceber a capacidade dos algoritmos de clusterização em subespaço de atuarem como filtros não-supervisionados de vetores de características. Neste momento, permanece o desafio da generalização do processo através da escolha do melhor cluster em um subconjunto da base de dados e a filtragem dos descritores da base completa pelo uso da aprendizagem supervisionada.

Tabela 4.8: Comparação do estado da arte da classificação da base *Oxford Buildings* com a metodologia de identificação de cenas baseada na filtragem de descritores usando clusterização em subespaço e posterior seleção manual do melhor cluster.

Conjunto de Busca	Melhor Resultado [Turcot & Lowe, 2009]	Filtragem de Descritores (% dos dados) 3,67	12,32
All Souls	0,661	0,580	**0,680**
Ashmolean	0,408	0,590	**0,740**
Balliol	0,160	0,510	**0,610**
Bodleian	0,292	0,430	**0,470**
Christ Church	0,414	**0,680**	**0,680**
Cornmarket	0,205	0,660	**0,780**
Hertford	0,554	0,570	**0,710**
Keble	0,352	**0,700**	**0,700**
Magdalen	0,024	0,340	**0,480**
Pitt Rivers	0,210	**0,690**	0,660
Radcliffe	**0,765**	0,690	0,750

4.5 Identificação de Cenas na base *Oxford Buildings*

Como os resultados preliminares dos testes para as bases de dados de Paris e Ouro Preto conseguiram incrementar a identificação de cenas usando apenas 1% da base de dados original, pretende-se aplicar a técnica de filtragem de descritores proposta neste trabalho, na base *Oxford Buildindgs*, para confrontar os resultados com aqueles obtidos por Turcot & Lowe [2009].

O protocolo experimental para a identificação das cenas é o mesmo descrito na Seção 4.1.1, usando o algoritmo E–MSSC. Em relação aos experimentos feitos por Philbin et al. [2007], e que serviram de base para os resultados obtidos por Turcot & Lowe [2009], foi realizada uma mudança no uso das imagens de busca, as quais passaram a ser usadas por completo, ao invés das regiões empregadas por Philbin et al. [2007]. Os resultados obtidos são mostrados na Tabela 4.8. Percebe-se que o uso do cluster representando 3,67% da base de dados possibilita o ganho da técnica ora apresentada em 9 dos 11 grupos de imagens, confrontados com os resultados de Turcot & Lowe [2009]. Por outro lado, ao se relaxar a restrição de se empregar o cluster de menor tamanho, há o cluster contendo 12,32% dos pontos da base que aumenta o ganho para 10 grupos em 11 possíveis.

Na próxima seção, são discutidos os resultados obtidos na identificação de cenas com o emprego da filtragem de descritores.

4.6 Análise dos Resultados

Foi possível observar que, nas três bases de dados testadas para a aplicação da identificação de cenas por filtragem de descritores, a aplicação dos algoritmos de clusterização em subespaço promoveu uma melhora significativa no ranque médio das imagens retornadas, em comparação com os resultados obtidos antes do uso da clusterização. Para a base de imagens de Paris, foi obtida uma taxa de melhoria de 59%. Em relação à base de dados de Ouro Preto, foi alcançado em torno de 43% de melhoria da identificação do conjunto de imagens de consulta proposto. Na base de dados de Oxford, apenas em um grupo de imagens de consulta os resultados da identificação de cenas por filtragem de descritores não foi melhor.

Entretanto, esses valores foram obtidos após a avaliação de cada cluster encontrado pelo algoritmo usado por Valle et al. [2009] e por Picard et al. [2009], ou seja, mesmo após intensa investigação, ainda não foi possível afirmar quais são as características que levam um cluster a ter uma melhor correspondência com as imagens de busca, ou não. Enquanto que na primeira base de dados, os pontos relativos a árvores e sombras prejudicaram o processo de busca das imagens, na segunda, a confusão pode ter sido causada pela ocorrência de estilos arquitetônicos muito similares para as várias construções presentes nas imagens e regiões altamente texturizadas. Esses resultados confirmam que, de fato, essa é uma tarefa muito desafiadora.

A experiência com a amostragem de dados no algoritmo FINDIT suscitou a modificação do algoritmo MSSC para melhorar seu desempenho em relação ao tempo de processamento, resultando no algoritmo E–MSSC. Sendo assim, o algoritmo E–MSSC foi implementado incorporando-se a técnica de amostragem proposta por Guha et al. [1998] e mostrou-se que sua performance na descoberta dos clusters é $96{,}41 \pm 3{,}1\%$ equivalente à do algoritmo original, compensada pelo ganho no tempo de execução, mesmo para grandes bases de dados, sendo cerca de 60 vezes mais rápido.

Adicionalmente, quando os algoritmos de clusterização em subespaço foram aplicados sobre as bases de dados, observou-se um efeito interessante dentro do espaço de descritores SIFT: alguns clusters, contendo uma pequena amostra dos dados, colaboraram para a correspondência entre imagens num grau muito maior que a base completa. Em comparação com Turcot & Lowe [2009], que mostram que 4% da base de dados é suficiente para obtenção do mesmo desempenho da base de dados completa, na tarefa

de identificação de cenas, este trabalho conseguiu melhorar a mesma tarefa usando algo em torno de 1% do universo total de descritores disponíveis. Entretanto, resta o estudo detalhado de todos os resultados obtidos para encontrar pistas importantes de quais são as características que tornam um cluster mais discriminativo que todos os dados disponíveis.

Seria interessante, ainda, a identificação automática dos clusters viáveis, isto é, aqueles que melhoram o casamento entre imagens, sem a necessidade do uso da inspeção exaustiva nos mesmos. O emprego de um algoritmo alternativo de casamento entre pontos, como a Transformada de Hough, proposta por Lowe [2004], é também uma possibilidade, assim como a seleção de descritores úteis proposta por Turcot & Lowe [2009].

4.7 Considerações

Neste capítulo, foi apresentado o protocolo experimental usado na identificação de cenas, aplicado em bases de imagens contendo fachadas de cidades históricas.

Em seguida, os algoritmos FINDIT [Woo et al., 2004] e MSSC [Gan et al., 2007] foram comparados empiricamente, através de uma base sintética multidimensional, e aplicados na filtragem de descritores das bases de imagens de Paris e Ouro Preto.

Ainda para a identificação de cenas, o novo algoritmo E–MSSC foi também comparado com o algoritmo MSSC na descoberta de clusters através de um conjunto de 162 bases sintéticas e, posteriormente, empregado na filtragem das duas bases de imagens, tendo sido evidenciada sua vantagem no tempo de processamento, enquanto manteve um índice de identificação próximo ao alcançado com a filtragem pelos outros dois algoritmos. A melhoria do processo de identificação de cenas foi experimentado, também, com o emprego do algoritmo E–MSSC na base *Oxford Buildings*.

Parte II

Classificação de Imagens

Capítulo 5

Trabalhos Relacionados

A segunda parte deste trabalho aborda a classificação de imagens, que emprega recursos observados na primeira parte, mas que possui, também, um arcabouço próprio e que tem evoluído muito desde a década passada.

Atualmente, não se pode pensar na classificação de imagens sem o emprego das técnicas de representação de imagens baseadas em dicionários visuais, considerando os vários trabalhos na literatura que abordam a classificação de imagens, baseados no emprego dessas técnicas. Não pretendemos oferecer ao leitor uma revisão detalhada de todos os trabalhos existentes, o que seria impossível devido à enorme ebulição que tem ocorrido nos últimos anos, mas, sim, explorar os principais expoentes e mais, aqueles relacionados com a classificação de fachadas.

Como contraponto, oferecemos uma reflexão sobre a abordagem baseada na análise dos *Nearest-Neighbors* (NN) para a classificação, ao invés do *Support Vector Machine* (SVM), eliminando-se, assim, a necessidade da etapa de treinamento.

A seguir, são discutidas algumas dessas principais técnicas e trabalhos, que se relacionam diretamente com as contribuições propostas para a classificação de imagens e que são apresentadas no próximo capítulo.

5.1 Métodos de Representação de Imagens Baseados em Dicionários Visuais

Esta seção detalha dois métodos de representação de imagens por dicionários visuais: *Bag-Of-Words* (BoW) [Sivic & Zisserman, 2003; Csurka et al., 2004] e *Spatial Pyramids Matching* (SPM) [Lazebnik et al., 2006]. Esses métodos foram escolhidos entre os diversos métodos empregados na literatura por sua importância para o desenvolvimento

deste trabalho. O método BoW [Sivic & Zisserman, 2003; Csurka et al., 2004] traz uma abordagem importante tanto para a identificação de cenas e objetos quanto para a classificação de imagens. Nele, o espaço de características locais é quantizado e o histograma da ocorrência das características locais presentes na imagem é usado para caracterizá-la como um único vetor de características. Já no método de casamento de características locais por Pirâmides Espaciais (*Spatial Pyramids Matching* (SPM)) [Lazebnik et al., 2006], o método BoW é estendido pela divisão hierárquica da imagem como requisito prévio para a construção das representações.

5.1.1 Bags-of-Words (BoW)

O método BoW foi proposto inicialmente por Sivic & Zisserman [2003] ao adaptar os conceitos de recuperação de documentos textuais para imagens e vídeos. Assim, ao invés de serem contabilizadas as repetições de vocábulos em um documento de forma a definir sua assinatura, passou-se a contabilizar a ocorrência de descritores similares em uma imagem, chamados de palavras visuais, correspondendo à assinatura visual desta. A forma de comparação entre as assinaturas das imagens manteve o mesmo conceito da recuperação de documentos, calculando-se a importância de cada palavra visual em uma determinada imagem através da Equação 5.1.

$$t_i = \frac{n_{id}}{n_d} log \frac{N}{n_i} \tag{5.1}$$

em que n_{id} representa o número de ocorrências da palavra visual i na imagem d, n_d, a quantidade de palavras visuais existentes na imagem d, n_i, a quantidade de ocorrências desta palavra visual na base de dados e N, o número de imagens da base de dados.

Essa abordagem de avaliação dos "documentos visuais" difere da solução proposta por Csurka et al. [2004], que emprega métodos estatísticos de aprendizagem para a categorização visual das imagens. Os autores propõem o uso de um conjunto de imagens de treinamento para geração de um modelo de classificação usando aprendizagem supervisionada multiclasses através do método Naïve Bayes e da Máquina de Vetores de Suporte (SVM). Sendo assim, as assinaturas geradas pelas palavras visuais para um subconjunto das imagens da base são usadas como entrada para os classificadores que, de posse das assinaturas do restante das imagens, faz a categorização visual das mesmas. Nos experimentos realizados, Csurka et al. [2004] obtiveram os melhores resultados ao empregarem o SVM.

Apesar de os objetivos de recuperação visual descritos por Sivic & Zisserman [2003] e por Csurka et al. [2004] serem distintos (o primeiro almeja a identificação de

objetos, o segundo, a classificação de imagens), as duas soluções geram os dicionários visuais da mesma maneira. Os descritores das imagens são clusterizados empregando-se o algoritmo K–means [MacQueen, 1967] e os centroides, ou seja, os pontos médios dos clusters encontrados, são as palavras visuais que formam o dicionário visual. Em seguida, a representação da imagem é criada através de um histograma de características visuais no qual cada intervalo contabiliza a resposta das características visuais da imagem a cada palavra visual.

Trabalhos recentes têm substituído a escolha dos dicionários visuais através do algoritmo K–means, pela seleção aleatória de centroides da base de dados, cuja quantidade varia de acordo com a natureza dos dados [Kläser et al., 2010; Perronnin et al., 2010; Avila et al., 2011]. Tal substituição tem se mostrado eficiente, economizando tempo de processamento, e, ao mesmo tempo, eficaz, mantendo e até mesmo aumentando os índices de acerto na classificação de imagens dos métodos baseados na representação de imagens por dicionários visuais.

Formalmente, a representação de imagens através do método BoW pode ser descrita como um processo de duas fases em que, na primeira, ocorre a codificação dos descritores da imagem segundo o dicionário visual escolhido e, na segunda, a contagem das respostas dos descritores a cada palavra do dicionário [Boureau et al., 2010]. Sendo assim, a codificação pode ser representada pela Equação 5.2 [Boureau et al., 2010; Avila et al., 2011].

$$\begin{aligned} &f : \mathbb{R}^n \rightarrow \mathbb{R}^k \\ &f(\vec{v}) = [f_{d_1}(\vec{v})\ f_{d_2}(\vec{v})\ ...\ f_{d_k}(\vec{v})] \\ &f_{d_j}(\vec{v}) \in \{0,1\}, f_{d_j}(\vec{v}) = 1 \iff j = \underset{1 \leq j \leq k}{\operatorname{argmin}} ||v - d_j||_2^2 \end{aligned} \tag{5.2}$$

em que n representa o número de dimensões dos vetores de características, k é o tamanho do dicionário visual e d_j representa a j-ésima palavra do dicionário.

Seguindo a notação anterior, uma das estratégias para a contagem final das respostas dos descritores às palavras do dicionário visual é realizada através da resposta média dada pela Equação 5.3 [Boureau et al., 2010].

$$g(\hat{I}) = \frac{\sum\limits_{i \in \hat{I}} f(\vec{v_i})}{|\hat{I}|} \tag{5.3}$$

em que $|\hat{I}|$ representa o número de vetores de características da imagem $\hat{I}$.

A codificação proposta pela Equação 5.2 perfaz uma associação rígida (*hard assignment*) entre o vetor de características e a palavra do dicionário visual. Entretanto,

trabalhos da literatura têm obtido bons resultados empregando a associação suave (*soft assignment*), na qual cada vetor de características tem associado a ele uma probabilidade de similaridade com cada uma das palavras do dicionário visual [Avila et al., 2011].

5.1.2 Casamento por Pirâmides Espaciais (SPM)

No modelo clássico do BoW, a informação espacial a respeito da imagem é perdida, ou seja, a assinatura produzida para a imagem não leva em consideração a disposição dos descritores locais na cena. Por outro lado, o Casamento por Pirâmides Espaciais (*Spatial Pyramids Matching* (SPM)) [Lazebnik et al., 2006] tem sido proposto para atenuar essa perda de informação espacial.

As Pirâmides Espaciais funcionam como um BoW multiníveis e têm sido usadas para incrementar os resultados em tarefas de classificação. Sua ideia é representar uma imagem através de um vetor, o qual inclui histogramas BoW construídos para repetidas divisões hierárquicas da imagem. Em seguida, as representações para as diversas divisões são concatenadas de forma ponderada (ou seja, cada nível tem um peso na representação) e o vetor final normalizado. No artigo introdutório do SPM, os autores comparam, ainda, a eficácia da técnica empregando "características fracas" e "características fortes". As primeiras são os descritores de imagens extraídos da forma convencional, de maneira que os pontos de interesse são escolhidos de acordo com sua relevância dentro da vizinhança, conforme visto na Seção 2.1, e para obtenção do segundo grupo de características os autores propõem a sua extração segundo uma grade fixa, sendo descrito um ponto a cada n na vertical e na horizontal. Sendo assim, $n \times n$ representa a densidade da grade. Os resultados obtidos por Lazebnik et al. [2006] mostram a superioridade das "características fortes" para bases de dados como Caltech–101 e Graz.

Para a fase de treinamento e classificação das imagens representadas pelo método SPM, é proposto por Lazebnik et al. [2006] o uso de um *kernel* para o casamento das pirâmides, dado pela Equação 5.4. Porém, trabalhos recentes têm empregado com sucesso classificadores SVM lineares que possuem a grande vantagem da escalabilidade sobre os vetores de características [Boureau et al., 2010; Perronnin et al., 2010; Avila et al., 2011].

$$\kappa^L(X, Y) = \frac{1}{2^L}\mathcal{I}^0 + \sum_{\ell=1}^{L} \frac{1}{2^{L-\ell+1}}\mathcal{I}^\ell \tag{5.4}$$

O funcionamento tradicional da técnica de codificação por Pirâmides Espaciais pode ser visto na Figura 5.1. São apresentados, na figura, os três níveis de criação das pirâmides, separados em três blocos. Na parte inferior de cada um dos blocos, pode ser visualizada a ponderação adotada: 1/4 para os níveis 0 e 1, e 1/2 para o nível 2 —, bem como as três palavras visuais empregadas para a quantização dos descritores das imagens: •, ◇ e +. Sendo assim, a ocorrência de cada um dos vocábulos visuais é tabulada por região e ponderada pelo nível ao qual pertence. A concatenação dos valores obtidos para cada nível forma, então, a assinatura da imagem.

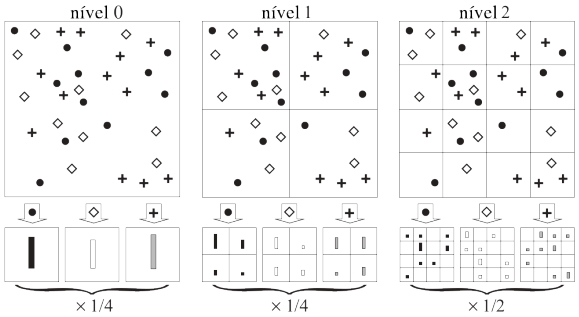


Figura 5.1: Diagrama extraído e adaptado de Lazebnik et al. [2006] que exemplifica a construção da representação por Pirâmides Espaciais com três níveis, respeitando o peso de cada um dos níveis.

Descritores de imagem como SIFT e SURF são considerados descritores de baixo nível, enquanto que as representações por dicionários visuais ganham status de descritores de nível médio [Boureau et al., 2010]. Qualquer um dos dois níveis de descritores pode ser empregado na tarefa de classificação de imagens de duas maneiras: a distância entre os descritores é calculada através de medidas de distância, como a Euclidiana, e os grupos com menor distância geral entre seus descritores são identificados; ou os descritores são submetidos a ferramentas de aprendizagem estatística, responsáveis por separar os grupos de descritores com características semelhantes.

5.2 Máquinas de Vetores de Suporte (*Support Vector Machines*)

A ferramenta de aprendizagem estatística utilizada em grande parte dos trabalhos sobre identificação de cenas e classificação de imagens é a Máquina de Vetores de Suporte (*Support Vector Machine* (SVM)) [Cortes & Vapnik, 1995; Vapnik, 1998]. O SVM é um classificador que tem por objetivo a construção de um hiperplano que separe dois conjuntos de dados. Como existem vários hiperplanos que cumprem esta função,

deseja-se aquele cuja distância mínima para os dados de ambos conjuntos seja a maior possível. Essa distância, chamada margem, é medida entre o hiperplano e os vetores de dados, de cada conjunto, mais próximos a ele, chamados de vetores de suporte (Figura 5.2).

Inicialmente, o SVM foi concebido como classificador de margem máxima, mas para certos conjuntos de dados pode não ser possível uma separação rígida entre eles. Para resolver essa questão, Cortes & Vapnik [1995] propuseram o uso de uma margem suave (*soft margin*), através da qual se tenta obter a divisão que melhor classifique os dados, sendo admitido algum erro de separação por parte do hiperplano.

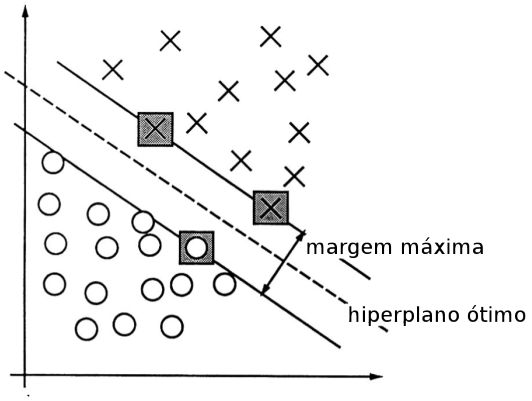


Figura 5.2: Exemplo extraído e adaptado de Cortes & Vapnik [1995]: separação de dois conjuntos em duas dimensões. Os pontos marcados em cinza representam os vetores de suporte usados para definir a maior distância entre as duas classes.

A aplicação do SVM sobre uma base de dados envolve a escolha de uma amostra dos dados como conjunto de treinamento para o qual se sabe, *a priori*, a classificação de cada elemento, que no caso de duas classes é $[-1,1]$. O conjunto de treinamento é empregado, então, para identificação dos vetores de suporte que definem o modelo de classificação dessa base de dados, sendo este usado para a predição da classificação do restante dos dados. O caso de duas classes pode ser generalizado para situações multi-classes.

O SVM exige a definição de uma série de parâmetros sensíveis à natureza dos dados e que estão vinculados ao *kernel* utilizado para a classificação, sendo comuns, na implementação do LIBSVM [Chang & Lin, 2011], usada neste trabalho, os *kernels* linear, polinomial, gaussiano e sigmoide. Sendo um classificador de margem suave (*soft margin*), um dos parâmetros que merecem destaque é o parâmetro de custo da margem C, que permite estabelecer o quão flexível será a separação entre os conjuntos de dados. Um valor baixo para C indica uma margem rígida, se aproximando do caso

do classificador de margem máxima, e um valor maior para C favorece a criação de um hiperplano de separação mais flexível.

Nos modelos paramétricos de classificação há o risco da ocorrência de *overfitting*, ou seja, uma flexibilização exagerada das margens de separação dos dados e, consequentemente, do hiperplano de classificação para o conjunto de treinamento, o que pode gerar um modelo de classificação ótimo para o conjunto de treinamento mas que falha para o restante da base de dados. O SVM teve grande sucesso na classificação de imagens, pois a complexidade do modelo aprendido não está relacionada com a dimensionalidade dos dados, e sim com o conceito teórico de "capacidade", ligado a um compromisso entre o tamanho da margem e a quantidade de vetores de suporte necessários para definir o hiperplano de separação dos dados [Duda et al., 2001]. Isso faz com que o SVM seja mais resistente ao problema de *overfitting* do que outros métodos como redes neurais ou a regressão, mesmo ao lidar com espaços de característica de dimensionalidade muito alta [Duda et al., 2001].

Embora o classificador SVM trabalhe com a separação linear dos dados, a separação não-linear pode ser obtida utilizando o *kernel trick*, que é o mapeamento implícito dos dados para outros espaços (usualmente de maior dimensionalidade), chamados de espaços de característica. A separação não-linear no espaço de dados corresponde a uma separação linear no espaço de características.

Os métodos de representação baseados em dicionários visuais, em conjunto com o classificador SVM, permitem o entendimento dos trabalhos relacionados com a classificação de imagens, abordados na próxima seção.

5.3 Classificação de Imagens

O estado da arte na classificação de imagens apresenta uma boa parte das técnicas baseadas em descritores locais, seja para promover uma classificação empregando aprendizagem supervisionada, como o SVM [Csurka et al., 2004; Lazebnik et al., 2006; Boureau et al., 2010; Perronnin et al., 2010; Avila et al., 2011], ou classificadores baseados na vizinhança dos descritores, como o (NN) [Boiman et al., 2008].

A metodologia de classificação das imagens para as técnicas baseadas em aprendizagem supervisionada envolve a escolha de imagens para servirem de conjunto de treinamento do classificador. Essas imagens de treinamento têm extraídos, então, seus descritores locais, que logo em seguida são quantizados e codificados por algum dos métodos de representação por dicionários visuais, baseados em BoW ou SPM. As assinaturas das imagens de treinamento alimentam o classificador, normalmente o SVM,

que irá produzir um modelo de classificação responsável pela anotação automática do conjunto de teste.

No trabalho desenvolvido por Csurka et al. [2004] é vista a associação entre a representação de imagens por dicionários visuais e o uso de classificadores estatísticos. Mais tarde, Lazebnik et al. [2006] propõem o uso de Pirâmides Espaciais, cuja codificação é modificada por Boureau et al. [2010] para garantir a esparsidade dos descritores de imagem. Perronnin et al. [2010] têm como objetivo a extensão da representação de imagens por SPM empregando a quantização feita por Modelos de Mistura Gaussiana (*Gaussian Mixture Models* (GMM)), que é a aplicação do *Fisher kernel* (FK) na representação de imagens por dicionários visuais. Mais recentemente, Avila et al. [2011] desenvolvem um descritor de imagem mais compacto e robusto pela inclusão, na representação da imagem pelos BoW, da medida da distância entre as palavras visuais quantizadas e o vocabulário visual, além da aplicação da associação suave (*soft assignment*) entre descritores e palavras visuais, baseada na mesma medida.

Os trabalhos citados anteriormente e vários outros encontrados na literatura abordam a classificação automática de imagens em bases de dados com grande volume de dados. Essas bases de dados são disponibilizadas juntamente com anotações individuais para cada imagem, com o intuito de permitir que autores comparem seus métodos. As bases de dados citadas com maior frequência são Caltech 101 [Li et al., 2004], Caltech 256 [Griffin et al., 2007], Pascal VOC [Everingham et al., 2007] (edições anuais) e LabelMe! [Russell et al., 2007].

Diferente das técnicas BoW e SPM, uma metodologia para a classificação de cenas naturais usando a representação de imagens por descritores semânticos locais é proposta por Vogel & Schiele [2007]. A ideia central por trás desse trabalho é realizar a modelagem semântica com o objetivo de classificar regiões da imagem em conceitos como água, céu, pedras e folhagem. O êxito na classificação das regiões da imagem é alcançado pelo uso do classificador SVM e nove classes de conceito. Além disso, a performance superior da modelagem semântica sobre o emprego direto de características de baixo nível é abordada.

Na contramão do uso do SVM, Boiman et al. [2008] propõem o uso dos NN (*nearest neighbors*) para medir a similaridade entre imagens e, consequentemente, definir a associação entre imagens e classes. As vantagens reivindicadas pelos autores são o uso de um método não-parametrizado e o ganho de tempo de processamento, com a exclusão da etapa de treinamento. Por outro lado, é apontada a pouca capacidade de generalização desse modelo de classificação.

A qualidade da classificação pode ser aferida submetendo um conjunto de imagens de teste com anotação conhecida ou verdade-terrestre[1] , e medindo o acerto do classificador. Para que a capacidade de generalização do classificador seja corretamente aferida, é preciso rigoroso cuidado para que eventuais informações sobre este conjunto de teste não contaminem a fase de treinamento como, por exemplo, o uso de imagens similares nos dois conjuntos. Por outro lado, é sabido que a escolha do conjunto de treinamento influencia na qualidade da classificação. Uma das estratégias para fazer a avaliação de qualidade, evitando a contaminação entre treino e teste, mas aferindo a variabilidade induzida pela escolha do treino, é a validação cruzada. Nela, o conjunto de treinamento é dividido em k subconjuntos disjuntos — chamados dobras ou *folds* — tendo, aproximadamente, o mesmo número de elementos. Em seguida, $k-1$ desses subconjuntos são usados para treinamento do classificador e o subconjunto restante é usado para teste do modelo de classificação. O processo é repetido k vezes, alternando-se o subconjunto de validação da classificação. A acurácia da classificação pode ser então estimada ao longo dessas rodadas, tanto em termos de valor médio, quanto em termos de uma flutuação em torno dessa média. A validação cruzada também pode ser usada como um procedimento interno em cada rodada, para otimizar certos parâmetros da classificação, usando apenas o conjunto de treino: os hiperparâmetros do SVM são frequentemente otimizados dessa forma.

A validação cruzada em *folds* não é a única forma de avaliar a variabilidade da acurácia de classificação devida à escolha do conjunto de treinamento. Outros projetos experimentais que incluam variação dos conjuntos de treino e teste podem ser contemplados. Um projeto que se tornou popular é o 5×2-*cross validation*, em que cinco divisões aleatórias da base são geradas, cada uma delas separando metade dos dados para treino e metade para a base. Outros projetos também são possíveis, desde que mantenham dados de treino e teste rigorosamente estanques.

Apesar de a recuperação de informação visual em cenas urbanas estar recebendo grande atenção do meio científico, haja vista o desafio proposto pela Nokia [2009] como parte dos Grandes Desafios da conferência ACM Multimídia, a classificação de imagens avançou significativamente para bases generalistas, em detrimento de problemas mais específicos, como é o caso da classificação de fachadas em relação ao seu estilo arquitetônico, abordado na próxima seção.

[1] esse termo, do inglês *ground-truth*, é herdado de sensoreamento remoto e significa literalmente "a verdade tal como observada no solo".

5.3.1 Classificação de Imagens em Cidades Históricas

Nos trabalhos citados anteriormente, é possível perceber que em nenhum momento se aborda o assunto de detecção automática de imagens em cidades históricas ou classificação de estilos arquitetônicos e, tampouco, a existência de bases de dados anotadas para esse fim é relatada. Portanto, é corriqueiro que trabalhos relacionados com essa área tenham que prover e anotar as suas próprias bases de dados, que geralmente não são disponibilizadas, trazendo dificuldades para a comparação entre métodos. Gerar as próprias bases de dados anotadas foi a solução encontrada por Shalunts et al. [2011, 2012a,b] e Mathias et al. [2011] para testar suas metodologias, descritas a partir de agora.

O método proposto por Shalunts et al. [2011], baseado na técnica de BoW, classifica janelas de edificações em três estilos arquitetônicos, sendo eles: Românico, Gótico e Renascentista/Barroco. Shalunts et al. [2011] avaliam a classificação final que será dada a uma janela levando em consideração a resposta máxima observada no histograma dos BoW feito a partir de sua imagem, considerando os três estilos citados anteriormente.

Ao invés de janelas, Shalunts et al. [2012a] utiliza domos, classificando os mesmos entre os estilos Renascentista, Russo e Islâmico. Para isso, esse método de classificação segue uma abordagem em três passos que determinam os estilos por eliminação: (i) a altura e a largura de um domo são usadas para separar os pertencentes ao estilo Islâmico; (ii) determinados tons da cor dourada ajudam a selecionar os domos Russos; e (iii) o passo final é baseado em BoW e detecta se o domo é Renascentista ou se pertence aos dois estilos anteriores e não foi classificado corretamente nos dois primeiros passos, usando novamente a resposta máxima do histograma dos BoW da imagem em relação aos estilos [Shalunts et al., 2011].

Adicionalmente, Shalunts et al. [2012b] classificam elementos arquitetônicos diferentes, estendendo Shalunts et al. [2011]. Os elementos são traceria[2], frontão[3] e balaustrada[4], associados ao estilo barroco ou ao gótico.

Indo além da classificação de elementos arquitetônicos, Mathias et al. [2011] abordam a identificação de estilos arquitetônicos em fachadas de edificações. Os autores propõem, inicialmente, a identificação da cena conforme sua natureza, como por exemplo: Sem edificações, Parte de edificação, Rua ou Fachadas. Caso a classe Fachadas seja identificada, a imagem é retificada e as fachadas presentes nela são separadas. Final-

[2]Ornamento feito em pedra.

[3]Arremate superior de portas e janelas, bem como elemento de vedação entre o telhado duas águas e as paredes.

[4]Conjunto de pequenas colunas presentes em corrimãos e guarda-corpos.

mente, o classificador *Naïve-Bayes Nearest-Neighbor* (NBNN) faz a distinção de cada fachada entre os estilos: Renascimento Flamenco[5], Haussmanniano[6] e Neoclássico.

Em um trabalho mais recente, Doersch et al. [2012] apresentam a associação entre imagens georreferenciadas e a detecção de elementos arquitetônicos de forma a possibilitar a classificação de imagens conforme sua localização geográfica. Dessa forma, cenas urbanas não anotadas podem ser relacionadas com as cidades segundo suas características arquitetônicas. No trabalho, são desenvolvidos vários testes e avaliadas as respostas para as cidades de Paris, Londres, Praga, Milão e Barcelona.

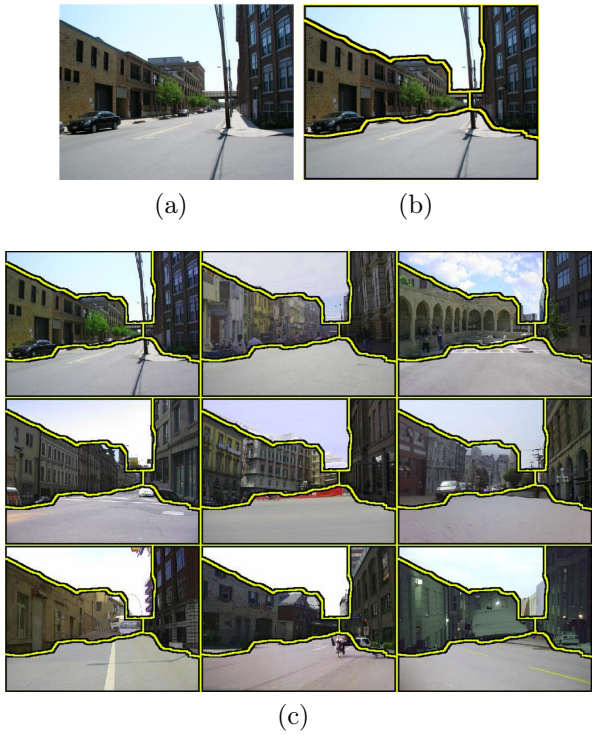
5.3.2 O Uso da Segmentação na Classificação de Imagens

A segmentação das cenas também pode contribuir para a identificação e classificação de objetos e imagens. São discutidos, a seguir, vários modelos de segmentação que, num primeiro momento, buscam a identificação de objetos específicos nas cenas para posteriormente segmentá-las conforme as informações obtidas inicialmente. Observa-se, também, o emprego da segmentação nos trabalhos sobre classificação de estilos arquitetônicos em cidades históricas citados na seção anterior.

Russell et al. [2009] tornam possível que uma imagem de busca seja descrita através de casamentos parciais de cenas similares. A otimização na técnica de casamentos parciais acontece através do emprego da segmentação baseada em MRF, que atua sobre as informações de bordas. Para cada imagem de busca é associada uma pilha contendo imagens semelhantes à primeira, usando o descritor Gist [Oliva & Torralba, 2001]. A pilha é usada para segmentar a imagem de busca através da detecção de bordas e, posteriormente, a detecção de regiões. A primeira é feita através do uso do algoritmo de detecção de bordas *Probability of Boundary Edge Detector* (PB) [Martin et al., 2004] e das imagens contidas na pilha, que auxiliam na descoberta das bordas entre objetos, que são de interesse para a segmentação, e as bordas internas aos objetos, que são descartadas. No caso das regiões, retalhos retangulares da pilha são comparados com os da imagem de busca e depois agrupados, usando o algoritmo *K–means*, formando regiões maiores na imagem. Finalmente, as duas informações são combinadas de forma a definir a segmentação definitiva da imagem. Os resultados obtidos por Russell et al. [2009] podem ser vistos na Figura 5.3. Os autores afirmam que, além do casamento entre cenas, a técnica pode ser usada em tarefas como reconhecimento de objetos ou computação gráfica.

[5]Movimento renascentista ocorrido nos Países Baixos.

[6]Decorrente da reforma urbana de Paris promovida por Georges-Eugène Haussmann de 1852 a 1870



(a) (b)

(c)

Figura 5.3: Exemplo do uso da segmentação usando composição de imagens extraído de Russell et al. [2009]: (a) apresenta a imagem de pesquisa usada, (b) a segmentação da imagem de pesquisa e (c) os resultados obtidos na busca usando a segmentação feita em (b).

A segmentação semântica de cenas urbanas também é abordada por Zhang et al. [2010] como um tópico de pesquisa importante para compreensão de cena (*scene understanding*) e modelagem baseada em imagens para cidades e áreas urbanas. A correspondência entre elementos para prover a transferência das anotações para a imagem de busca é baseada na técnica *K-Nearest-Neighbors–Markov Random Field* (KNN–MRF). É importante frisar que os elementos que serão combinados entre as imagens são chamados de *superpixel*, ou seja, são concatenações de grupos de pixels vizinhos nas imagens.

O uso da técnica BoW por Sivic et al. [2005] objetiva descobrir categorias de objetos em um conjunto de imagens não anotadas e, consequentemente, segmentar essas imagens. A aplicação de *doublets*, que codificam regiões espacialmente sobrepostas e são uma extensão do vocabulário habitual da técnica BoW, propicia uma segmentação mais refinada das imagens, sendo que a aprendizagem não-supervisionada sobre as divisões se dá pelo algoritmo *probabilistic Latent Semantic Analysis* (pLSA) e é comparada com resultados obtidos através do algoritmo *K–means*.

Nos trabalhos de classificação de imagens de cidades históricas é muito comum, também, o uso da segmentação, uma vez que o processo de classificação é dirigido para elementos arquitetônicos específicos, presentes nas edificações.

No trabalho de Shalunts et al. [2011], cujo objetivo é a detecção do estilo arquitetônico de janelas, os detectores de janelas vistos por Ali et al. [2007]; Recky & Leberl [2010a,b] são usados para a geração de delimitadores retangulares em volta dos objetos. Por outro lado, Shalunts et al. [2012a] propõem que as regiões delimitadoras dos domos a serem classificados sejam geradas manualmente, através da intervenção de usuários.

Outras propostas de solução do problema de segmentação semântica automática de imagens envolvem o uso de um repositório de imagens anotadas para realizar a transferência das anotações para os objetos, a partir de uma segmentação inicial [Liu et al., 2009] e, ainda, há os que tem o objetivo de localizar a posição dos objetos dentro da imagem para uma posterior reconstrução tridimensional [Snavely et al., 2006].

5.4 Considerações

Foram vistas, aqui, algumas das principais técnicas utilizadas nas tarefas de classificação de imagens. Primeiramente, as técnicas de representação de imagens por meio de dicionários visuais foram abordadas, sendo discutidas as suas principais vantagens e, também, seus pontos fracos.

Posteriormente, deu-se atenção a uma das principais técnicas de aprendizagem supervisionada voltada para a classificação de imagens que são as SVM.

O uso combinado das técnicas vistas tem proporcionado enormes avanços na classificação automática de imagens, mas carece, ainda, do uso de informações semânticas a respeito do conteúdo das imagens. Essa abordagem traz benefícios importantes, como uma maior robustez dos descritores de nível médio gerados em relação a transformações afins e melhor controle da compacidade desses descritores.

Os métodos de representação de imagens por dicionários visuais BoW [Sivic & Zisserman, 2003; Csurka et al., 2004] e SPM [Lazebnik et al., 2006] serão a base para o método *Semantic Spatial Pyramids* (SSP), cujo objetivo é efetuar a quantização dos descritores das imagens através de dicionários visuais e regiões semânticas. A ideia de empregar regiões semânticas na tarefa de classificação de imagens veio dos trabalhos propostos por Oliva & Torralba [2001] e Torralba et al. [2008], nos quais resultados promissores foram obtidos para segmentar regiões das imagens de *street-view* usando informação semântica. O sucesso desses autores nos motivou a questionar se regiões

semanticamente obtidas não poderiam ser *utilizadas* na melhoria das representações baseadas em BoW.

Finalmente, observam-se diversos trabalhos da literatura relacionados com as propostas presentes neste texto, como em Shalunts et al. [2011, 2012a,b] e Mathias et al. [2011], na classificação de estilos arquitetônicos em bases de dados de fachadas de cidades históricas.

Capítulo 6

Contribuições Propostas para a Classificação de Imagens

Observou-se, no capítulo anterior, o enorme desafio que envolve a tarefa de classificação de imagens, principalmente em bases de dados em que há uma grande diversidade de classes e quantidade inconstante de imagens de treinamento para cada uma delas.

Neste capítulo, é apresentada uma metodologia de classificação de imagens mediante a inclusão de informação semântica na construção da representação das imagens por dicionários visuais.

6.1 Classificação de Imagens

Na classificação de imagens, dado um conjunto de imagens $I = \{\hat{I}_1, \hat{I}_2, ..., \hat{I}_N\}$ e um conjunto de classes $C = \{C_1, C_2, ..., C_k\}$, deseja-se associar cada elemento em I a uma classe em C, minimizando-se o erro durante a realização dessa associação. Na associação rígida de classes, a probabilidade $P(\hat{I}_j \in C_k)$ de uma imagem $\hat{I}_j$ pertencer a uma classe C_k é dada por

$$P(\hat{I}_j \in C_k) = 1 \iff k = \underset{C_k \in C}{\operatorname{argmin}}\, dist(\hat{I}_j, C_k) \tag{6.1}$$

Segundo a Equação 6.1, uma imagem I será associada com uma classe de imagens C_k que estiver mais próxima dela, dentre todas as outras classes possíveis e de acordo com a métrica escolhida. Entretanto, em um espaço multidimensional, como é o caso dos descritores multimídia, a relação de distância entre objetos é muito sensível. Além disso, arbitrar um representante da classe para o qual será medida a distância pode não ser uma boa ideia devido a questões relativas à variabilidade interna da classe. De

qualquer forma, o emprego da força bruta para medir a distância de cada imagem I que se deseja classificar a cada representante de uma classe se torna impraticável num contexto de milhares de imagens e de descritores com várias dimensões.

Nesse caso, a solução apresentada pela literatura é o emprego métodos de otimização, como o SVM, que são utilizados para realizar a classificação sem a necessidade de confrontar cada elemento com cada uma das possibilidades disponíveis.

Outro fator importante é a escolha do método de representação das imagens, que pode contribuir para tornar mais preciso o processo de classificação de acordo com as informações que são associadas à representação.

Para a classificação de imagens, o que se usa atualmente são os métodos de representação através de dicionários visuais, apresentados inicialmente por Sivic & Zisserman [2003] e Csurka et al. [2004]. Entretanto, esses métodos carecem de informação espacial sobre a disposição dos objetos [Lazebnik et al., 2006], e também deixam de lado informações sobre o significado semântico desses objetos. Do ponto de vista da informação espacial, o assunto tem sido explorado constantemente na literatura, sendo provada a sua utilidade [Lazebnik et al., 2006; Boureau et al., 2010; Perronnin et al., 2010; Avila et al., 2011]. Em contrapartida, a informação semântica agregada aos objetos ainda é pouco abordada [Russell et al., 2009; Zhang et al., 2010], mas pretendemos mostrar adiante que ela é de vital importância no avanço do estado da arte da classificação de imagens.

6.2 Método Pirâmides Espaciais Semânticas

A técnica SPM, vista na Seção 5.1.2, atenua a falta de informação espacial na representação por dicionários visuais, numa tentativa de relacionar a ocorrência de descritores locais com determinadas regiões da imagem, a partir da representação global feita pelos BoW, o que tem resultado em avanços na classificação automática de imagens em grandes bases de dados [Perronnin et al., 2010]. Por outro lado, a associação de regiões da imagem com conceitos semânticos produz bons resultados na identificação e classificação de imagens, como foi discutido na Seção 5.3.2. Esta seção pretende, então, delinear a união das duas técnicas: SPM e segmentação semântica de imagens.

A principal diferença do método proposto para o SPM tradicional é a forma como os níveis da pirâmide de BoWs são construídos: na SPM tradicional, usam-se níveis hierárquicos fixos, que não são adaptados aos dados e não estão necessariamente relacionados ao conteúdo da imagem. Dessa forma, nos trabalhos encontrados na literatura ocorre, no máximo, a alteração das divisões feitas, ou seja, ao invés da divisão tradici-

onal dos níveis da pirâmide (imagem inteira no primeiro nível, quadrantes de mesmo tamanho no segundo nível e reparticionamento dos quadrantes em novos quadrantes no terceiro nível) [Lazebnik et al., 2006], adotam-se os dois primeiros níveis originais e o terceiro nível com três divisões horizontais ou verticais da imagem [Avila et al., 2011].

A técnica de Pirâmides Espaciais Semânticas ou SSP, por sua vez, relaxa essas restrições, possibilitando uma melhor adequação das regiões à aplicação alvo. Além disso, como as regiões passam a ser escolhidas de acordo com o objeto que representam na imagem, elas passam a ter um significado semântico. Sendo assim, cada objeto de interesse deve ser delimitado por uma região e, de acordo com a quantidade de ocorrências de cada um desses objetos dentro da imagem, obtém-se uma representação final que naturalmente terá um peso maior para objetos de maior frequência.

Na prática, a técnica consiste na definição de áreas retangulares representando as regiões de interesse na imagem, que recebem rótulos de acordo com os objetos delimitados. Dessa forma, cada imagem tem um mapa associado a ela que será usado na construção das SSP.

A segmentação e anotação dos objetos pode ser feita manualmente, como sugerido por Shalunts et al. [2012a], empregando-se alguma técnica de segmentação automática, como proposto por Sivic et al. [2005] e Zhang et al. [2010], para posterior anotação por usuários ou, ainda, usando em conjunto técnicas de segmentação de imagens e transferência de anotações para anotação automática dos objetos identificados [Snavely et al., 2006; Liu et al., 2009]. Entretanto, sabe-se que qualquer uma das três técnicas está sujeita a erros, visto que a primeira envolve a ação direta de usuários, a segunda prevê uma segmentação automática das cenas visando uma reconstrução tridimensional, permitindo ao usuário realizar anotações em regiões das imagens que são transferidas para outras imagens onde as regiões estão presentes [Snavely et al., 2006], e a terceira é baseada na segmentação e anotações automáticas e, apesar de usar um modelo não paramétrico de análise da cena, falha ao se deparar com objetos para os quais as amostras de treino são escassas [Liu et al., 2009].

Na construção da representação da imagem, ao invés de se usar a divisão em níveis hierárquicos, a técnica de Pirâmides Espaciais Semânticas emprega as regiões de interesse detectadas para construir os vetores de características de nível intermediário que irão representar as imagens. Para isso, são montados histogramas visuais para cada tipo de objeto presente na base de dados e detectado na imagem, usando-se a soma das ocorrências no caso de repetições do mesmo objeto. A concatenação dos histogramas do objeto para formar a representação da imagem ocorre de forma direta, sem ponderação, devendo-se manter a ordem da concatenação para todas as imagens, garantindo-se, com isso, que objetos de mesma natureza sejam comparados, a exemplo

Entrada:
DB : base de dados de descritores,
D_{img} : descritores da imagem,
R_{img} : regiões detectadas na imagem,
R_{tipo} : tipos de regiões,
Dic : dicionário visual
Saída: Rep_{img} : vetor $1 \times (|Dic| \times |R|)$ representando a imagem

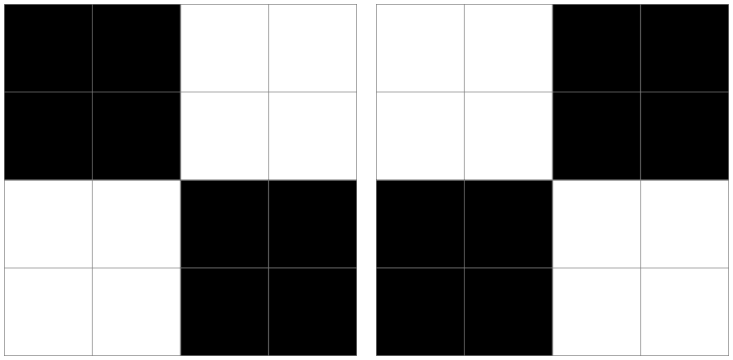
```
1  //Quantização dos descritores por região.
2  para todo r ∈ R_img faça
3  |  h_r ← CriarHistograma(r, DB, Dic);
4  |  //Regularização dos histogramas pela aplicação da média sobre a região.
5  |  //|r| denota o número de pontos dentro da região.
6  |  h_r ← h_r/|r|;
7  fim
8  //União das regiões de mesmo tipo.
9  para todo r_tipo ∈ R_tipo faça
10 |  para todo h_r ∈ R_img faça
11 |  |  se Tipo(h_r) == r_tipo então
12 |  |  |  h_{r_tipo} = h_{r_tipo} + h_r
13 |  |  fim
14 |  fim
15 fim
16 //Concatenação dos histogramas de cada região.
17 para todo r_tipo ∈ R_tipo faça
18 |  Concatena(Rep_img, h_{r_tipo});
19 fim
20 retorna Rep_img
```

Algoritmo 2: Pirâmides Espacial Semânticas

do que ocorre com as regiões hierárquicas do método SPM [Lazebnik et al., 2006]. Após a concatenação, o vetor final que representa a imagem é normalizado, empregando-se a norma ℓ_1. Essa montagem semântica da pirâmide gera representações esparsas das imagens, o que contribui para o aumento das taxas de classificação [Boureau et al., 2010].

Como forma de ilustrar a montagem das pirâmides semânticas e sua comparação com o método SPM, vamos usar um pequeno exemplo onde a imagem de treinamento é formada por 16 pixels e duas classes, como visto na Figura 6.1a, e sua imagem correspondente, usada para teste, é proveniente do padrão original rotacionado em 90°, como visto na Figura 6.1b.

Sem prejuízo da generalidade, vamos considerar o descritor de cada pixel como sendo o valor correspondente à cor presente no pixel, nesse caso 0 para a cor preta e



(a) (b)

Figura 6.1: Exemplo da comparação da representação por dicionários visuais do padrão usado para treinamento, visto à esquerda, e da versão rotacionada em 90° do mesmo padrão, vista à direita

1 para a cor branca, e o dicionário visual como sendo $Dic = \{0,1\}$. Sendo assim, as regiões consideradas para a montagem das SSP serão os quadrados brancos e pretos, formados, cada um, por quatro pixels. Após a contagem da ocorrência das palavras visuais na Figura 6.1a, o descritores da imagem pelas SSP, sem normalização, serão dados por

$$\begin{cases} n_0 = \begin{bmatrix} 8 & 8 \end{bmatrix}, \\ n_{r_1} = \begin{bmatrix} 8 & 0 \end{bmatrix}, e \\ n_{r_2} = \begin{bmatrix} 0 & 8 \end{bmatrix}. \end{cases}$$

sendo o primeiro valor referente à ocorrência da palavra visual 1 e, consequentemente, o segundo valor referente à outra palavra visual considerada. Observa-se que a representação da Figura 6.1b pelo mesmo método será idêntica à primeira, ou seja, o descritor formado pela concatenação das regiões e posterior normalização, para as duas imagens da Figura 6.1 será $d = \begin{bmatrix} 0{,}25 & 0{,}25 & 0{,}25 & 0 & 0 & 0{,}25 \end{bmatrix}$.

Agora, considerando os mesmos descritores e dicionário visual, vamos elaborar a representação das mesmas imagens pelo método SPM. Inicialmente, para a Figura 6.1a, o descritores dos níveis serão dados por

$$\begin{cases} n_0 = \begin{bmatrix} 8 & 8 \end{bmatrix}, \\ n_1 = \begin{bmatrix} 4 & 0 & 0 & 4 & 4 & 0 & 0 & 4 \end{bmatrix}, e \\ n_2 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \end{cases}$$

O descritor final será a concatenação desses vetores, que nesse exemplo serão considerados sem normalização e sem ponderação. Já para a Figura 6.1b, os descritores

dos níveis das SPM passam a ser

$$\begin{cases} n_0 = \left[8\,8\right], \\ n_1 = \left[0\,4\,4\,0\,0\,4\,4\,0\right], e \\ n_2 = \left[0\,1\,0\,1\,0\,1\,0\,1\,1\,0\,1\,0\,1\,0\,0\,1\,0\,1\,0\,1\,0\,1\,1\,0\,1\,0\,1\,0\,1\,0\right]. \end{cases}$$

É possível perceber que, na representação das imagens da Figura 6.1 pelas tradicionais SPM, os descritores finais de cada uma das imagens, gerados pela concatenação dos níveis, são maiores que aqueles produzidos pela representação por SSP e, também, são diferentes. Ou seja, o simples fato de uma imagem estar rotacionada em relação à outra faz com que sejam gerados descritores distintos para as SPM, enquanto que os descritores gerados pelo novo método se mantêm os mesmos.

Em relação ao tamanho do vetor gerado pelas codificações SPM e SSP, ele está relacionado com as quantidades respectivas de divisões e regiões consideradas. No caso do exemplo da Figura 6.1, o vetor gerado pela codificação original proposta para o método SPM contém $1 + 4 + 16$ divisões, ao passo que, para o método SSP, são empregadas apenas duas regiões. Dessa forma, na codificação SSP cada imagem é associada a um vetor de características de tamanho $|Regiões| \times |Dicionário|$, sendo *Regiões* o conjunto de objetos ou regiões observadas para a codificação e *Dicionário* o conjunto de palavras visuais. Assim, independentemente da quantidade de ocorrências dos objetos dentro da imagem, o vetor de características tem sempre o mesmo tamanho, visto que todas as respostas encontradas para um determinado tipo de objeto são acumuladas em um único segmento do vetor completo, pelo somatório das respostas a cada palavra visual (Algoritmo 2).

Nos trabalhos sobre classificação de estilo arquitetônico discutidos na Seção 5.3.1, observa-se, predominantemente, a utilização da técnica BoW na codificação das imagens submetidas ao processo de classificação [Shalunts et al., 2011, 2012a,b]. Entretanto, há a aplicação, também, de modelos não-paramétricos de classificação, como o NBNN [Mathias et al., 2011]. Portanto, percebe-se que, até o momento, não há a utilização das pirâmides espaciais, SPM ou SSP, nesse tipo de aplicação, sendo uma originalidade deste trabalho. Principalmente no que diz repeito à proposta de uso das SSP, constata-se sua validade para os trabalhos citados anteriormente, uma vez que em alguns deles há a delimitação de regiões, como pré-processamento da classificação, a exemplo do que ocorre no protocolo discutido aqui.

O diagrama de funcionamento do método pode ser visto na Figura 6.2. Inicialmente, cada imagem do conjunto de treinamento tem seus descritores extraídos pelo

SIFT e a partir de todos os descritores gerados é selecionado aleatoriamente um certo número de descritores para servirem de dicionário visual. Paralelamente à seleção do dicionário visual, regiões de interesse são delimitadas nas imagens para auxiliar na quantização dos descritores. A assinatura de cada imagem começa a ser construída pela geração dos histogramas de ocorrência das palavras visuais dentro de cada região, sendo que as contagens para regiões semelhantes são somadas, posteriormente, em um único histograma para aquele tipo de região. Os histogramas de cada tipo de região são concatenados juntamente com o histograma da imagem completa (BoW tradicional), formando um único vetor para a imagem que é, em seguida, normalizado, finalizando a construção da assinatura da imagem. As assinaturas das imagens de treinamento são submetidas ao SVM para criar o modelo de treinamento que servirá para predizer a classificação de uma imagem de teste cuja assinatura é gerada conforme os passos citados anteriormente.

Na Seção 7.3 o novo método Pirâmides Espaciais Semânticas é aplicado na classificação do estilo arquitetônico de fachadas de cidades históricas.

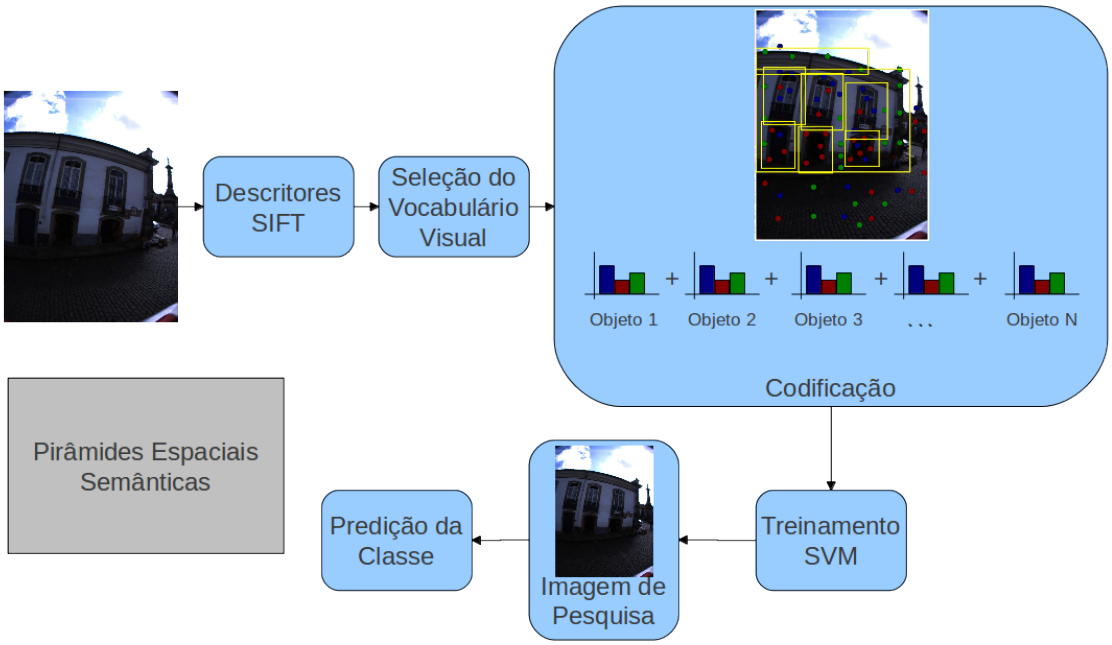


Figura 6.2: Novo Método Pirâmides Espaciais Semânticas

6.3 Considerações

Após a formalização da recuperação de informação visual na classificação de imagens, foram discutidas as principais contribuições deste trabalho.

Para a classificação de imagens, é apresentada uma nova forma de codificação dos descritores, baseada em regiões semânticas e dicionários visuais. A codificação SSP proposta inclui informações não utilizadas na técnica SPM [Lazebnik et al., 2006]. As principais vantagens do uso da nova técnica são uma maior robustez das representações geradas em relação a transformações afins nas imagens e a dependência dos vetores gerados em relação às regiões consideradas, ou seja, quanto menor a quantidade de regiões, mais compacto é o vetor. Caso contrário, um maior número de regiões leva a um vetor maior e, consequentemente, mais esparso.

Capítulo 7

Experimentos

Neste capítulo, é apresentado o protocolo experimental adotado para a classificação de estilos arquitetônicos empregando representações por SSP e SPM das imagens, em bases de dados de cenas urbanas. Ao final, os resultados obtidos são detalhados e discutidos.

7.1 Protocolo Experimental

A seguir, é descrito o critério empregado para definir as divisões das Pirâmides Espaciais (SPM) e das Pirâmides Semânticas (SSP), escolher os dicionários visuais empregados e selecionar as imagens para compor os conjuntos de treinamento e teste dos experimentos.

7.1.1 Classificação de Imagens por Estilos Arquitetônicos

A aplicação da classificação de estilos arquitetônicos em imagens foi realizada, inicialmente, através das Pirâmides Espaciais Semânticas (SSP). Primeiramente, as imagens são separadas em dois conjuntos distintos, chamados de conjunto de treinamento e teste. Em seguida, cada imagem foi descrita da forma convencionada na literatura relacionada à classificação de imagens, que prevê a escolha de uma grade fixa de pontos, aos quais são associados vetores de características. Sendo assim, foi aplicado o algoritmo SIFT, usando-se uma grade 8×8 de pixels.

A partir dos descritores gerados para as imagens do conjunto de treinamento, um dicionário visual é escolhido aleatoriamente. Nos experimentos realizados, a quantidade foi fixada em 4 000 descritores; quantidade esta empregada em diversos trabalhos da

literatura e que se mostra suficiente para grandes bases de dados, como as usadas por Kläser et al. [2010], Perronnin et al. [2010] e Avila et al. [2011].

Simultaneamente, pela intervenção de usuários, todas as imagens da base são avaliadas para a delimitação de regiões de interesse contendo os objetos considerados na classificação do estilo arquitetônico, que no caso deste trabalho são: porta, janela, porta-balcão, telhado e fachada (ver Seção 7.2).

Para a segmentação dos elementos arquitetônicos, as imagens da base de dados são distribuídas entre voluntários que são incumbidos de delimitar as regiões envolvendo os elementos arquitetônicos citados anteriormente. O software utilizado é o GIMP, que permite a geração de um código *HyperText Markup Language* (HTML) com a localização das regiões marcadas. Posteriormente, os arquivos HTML, um para cada imagem, são convertidos em um arquivo de texto contendo a quantidade de regiões detectadas, o número de elementos e, para cada região, sua identificação (porta, janela, porta-balcão, telhado, fachada) e sua localização dentro da imagem.

A codificação das imagens ocorre pela quantização dos descritores contidos dentro de cada região selecionada e posterior concatenação dos histogramas para cada um dos elementos arquitetônicos. Antes disso, regiões quantizadas para o mesmo elemento arquitetônico são representadas pela soma de seus histogramas. O passo final da codificação consiste na normalização do vetor usando a norma ℓ_1.

Os vetores gerados para o conjunto de treinamento são submetidos, então, ao classificador SVM, empregando-se a validação cruzada com cinco dobras ou divisões(*folds*) para se determinar o melhor valor para C (ver Seção 5.2), usando um *kernel* linear.

Finalmente, as imagens de teste são codificadas segundo o mesmo protocolo seguido para as imagens de treinamento e usando o mesmo dicionário visual. Os vetores gerados para cada imagem são submetidos ao classificador para a predição da classe a que pertencem e aferição da taxa de classificação.

Com o objetivo de validação da codificação proposta, a mesma classificação foi realizada por meio da representação das imagens pelo método SPM. Entretanto, para uma comparação mais justa entre os métodos, os descritores SIFT das imagens e o dicionário visual são os mesmos empregados na codificação pela técnica SSP. Cada imagem da base de treinamento é codificada para ser representada por um vetor construído a partir da quantização dos descritores segundo o dicionário visual escolhido e os seguintes níveis da pirâmide espacial: (i) a imagem completa (BoW tradicional [Lazebnik et al., 2006]); (ii) divisão da imagem em três faixas horizontais idênticas; e (iii) a imagem dividida em quatro partes iguais. O objetivo do segundo nível, diferente da divisão original apresentada por Lazebnik et al. [2006], é estabelecer, aproximadamente, as divisões entre céu, construção e nível da rua. Os segmentos de vetor construídos para

cada região são ponderados e posteriormente concatenados, conforme a técnica original. As etapas de treinamento e predição da classificação seguem as mesmas diretrizes do método SSP.

A próxima seção trata das principais características das fachadas barrocas, empregadas na avaliação da classificação SSP.

7.2 Barroco Mineiro

Nesta seção, expomos as principais características do estilo Barroco desenvolvido no estado de Minas Gerais, justificando a escolha dos elementos arquitetônicos empregados na classificação das imagens obtidas na cidade de Ouro Preto.

Segundo Ávila et al. [1996], o estilo arquitetônico Barroco, além de sua abrangência em outras áreas como música e literatura, confunde-se com a história brasileira e, principalmente, com a do estado de Minas Gerais. Ressalta-se a expressão desse estilo nas igrejas católicas presentes em cidades como Tiradentes, São João del Rei, Mariana e Ouro Preto mas, também, na constituição urbanística dessas cidades, onde é mantida, até os dias de hoje, com maior integridade e coerência, na cidade de Ouro Preto. Para isso, destacam-se a forte influência do ciclo do ouro e o fato dessa cidade ter sido capital do estado por um século.

No que tange as fachadas das edificações erigidas em estilo barroco, de Mello [1985] destaca sua horizontalidade, realçada por grandes beirais nos telhados e a distribuição equilibrada das janelas, no caso de casas construídas em lotes maiores, e a construção de sobrados, com fachadas coladas umas às outras, no caso de construções feitas em terrenos menores. Em relação aos sobrados, percebe-se a ocorrência das sacadas, ou balcões, e dos telhados arrematados de beirais com cimalhas. Algumas variações desses importantes elementos das fachadas barrocas podem ser vistas nas Figuras 7.1, 7.2, 7.3 e 7.4.

Sendo assim, optou-se pela seleção de cinco regiões para a codificação das imagens pela técnica SSP: porta, janela, porta-balcão, telhado e fachada. Os resultados dos experimentos podem ser vistos na próxima seção.

7.3 Resultados da Classificação de Estilos Arquitetônicos

Na classificação dos estilos arquitetônicos para a base de dados de Ouro Preto, notadamente, neste experimento, os estilos barroco e moderno, empregaram-se as Pirâmides

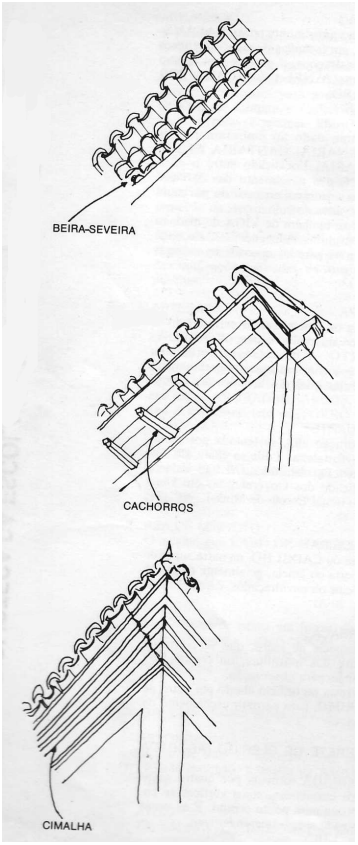


Figura 7.1: Acabamento típico dos telhados das residências barrocas. Extraído e adaptado de Ávila et al. [1996]

Espaciais (SPM) [Lazebnik et al., 2006] e as Pirâmides Espaciais Semânticas (SSP), sendo esta última proposta deste trabalho (Seção 6.2). Apresentamos, também, à guisa de *baseline*, o resultado utilizando os *bags of words* (BoW) simples, sem informação espacial.

Utilizamos uma base de 1 000 imagens anotadas a partir da base de imagens de Ouro Preto, que chamamos de OP1K. Para evitar a contaminação dos conjuntos de treino e teste, em todos os casos, os conjuntos foram amostrados de imagens das câmeras laterais em lados opostos (correspondendo às fachadas do lado par e do lado ímpar das ruas). Para avaliar a variabilidade devida à escolha das imagens, executamos cada experimento em 20 rodadas, selecionando, a cada rodada, dois conjuntos desbalanceados de treinamento e teste com, respectivamente 500 e 200 imagens.

Dada a restrição de se usar as câmeras laterais em lados opostos para a constituição dos conjuntos de treinamento e teste, para evitar a contaminação desses conjuntos, foi constatada a seguinte quantidade de imagens: Câmera 1 — 383 fachadas barrocas e 302 modernas —, e Câmera 2 — 117 fachadas barrocas e 198 modernas. Sendo assim,

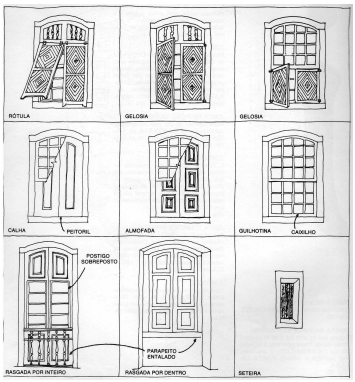


Figura 7.2: Exemplos de janelas e porta-balcão (canto inferior esquerdo) comuns nas construções barrocas. Extraído e adaptado de Ávila et al. [1996]

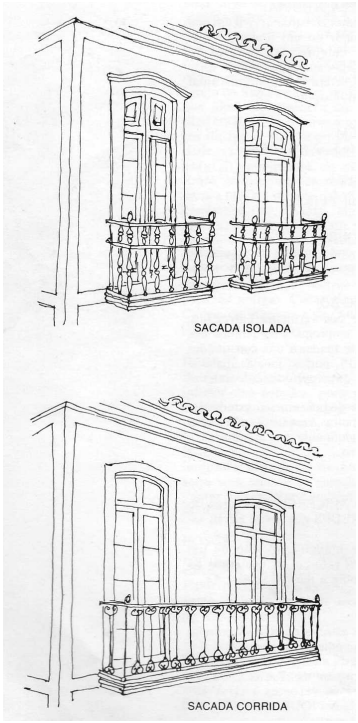


Figura 7.3: Exemplos de sacadas e balcões presentes nos sobrados do período colonial barroco. Extraído e adaptado de Ávila et al. [1996]

foram escolhidas, aleatoriamente, 250 fachadas barrocas e 250 modernas da Câmera 1 para composição do conjunto de treinamento, a cada rodada, e 100 fachadas barrocas e 100 modernas para o conjunto de teste, a cada rodada também.

Extraímos das imagens, convertidas para escala de cinza, os descritores SIFT e, a partir das imagens de treinamento em cada experimento, selecionamos um *codebook* de 4 000 descritores. O *codebook* foi então utilizado para criar a representação de nível-médio das palavras, em três modalidades:

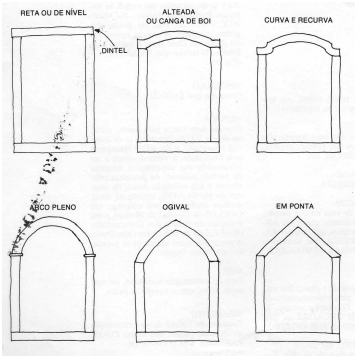


Figura 7.4: Estilos de acabamento de portas e janelas barrocas.. Extraído e adaptado de Ávila et al. [1996]

- *Bags of words* (BoW), obtidos por *hard-assignment* e *sum-pooling*. Isso corresponde a criar histogramas de contagem das palavras visuais, considerando cada descritor local como representado pela *codeword* mais próxima no *codebook*;

- Pirâmides Espaciais (SPM), obtidas criando um *bag of words* como acima, para 3 níveis de pirâmide: a imagem inteira, três faixas horizontais, e quatro quadrantes;

- Pirâmides Espaciais Semânticas (SSP), obtidas criando um *bag of words* como acima, para a imagem inteira e para as regiões correspondendo à: fachada, telhado, porta, janela e porta-balcão;

Ao final, todas as representações foram normalizadas usando a norma ℓ_1.

Os resultados dos experimentos podem ser conferidos na Tabela 7.1. No topo da tabela estão as acurácias médias de classificação das três técnicas, bem como os intervalos de confiança para $\alpha = 0{,}05$. Apesar de os intervalos de confiança se cruzarem, um teste-t pareado de uma cauda mostra que a diferença entre o SSP e o SPM é significativa ($p\text{-}value = 0{,}01$). A diferença entre o SSP e o BoW é, portanto, também significativa ($p\text{-}value = 0{,}0008$), mas a diferença entre o SPM e o BoW falha no teste de significância ($p\text{-}value = 0{,}20$).

A segunda parte da tabela explora como os resultados do SSP são afetados por perturbações induzidas na segmentação manual. Introduzimos dois tipos de perturbação: a remoção aleatória de regiões segmentadas (correspondendo ao erro de "não detecção" de regiões), e à introdução de um ruído Gaussiano nas coordenadas do retângulo que delimita a região (correspondendo a uma perda de precisão na delimitação correta da região). A deleção de 15% corresponde a deixar de detectar quase 1 região a cada 6, o que pode ter impacto severo na representação se a região perdida for a única daquele tipo na fachada (por exemplo, perda do telhado, ou perda da única porta de

uma fachada). Para o ruído Gaussiano, escolhemos um desvio padrão de 40 pixels, o que corresponde a uma flutuação bastante considerável nas coordenadas dos elementos, lembrando que o tamanho *total* da imagem é de 1024×768 pixels, e que as dimensões lineares típicas de um elemento como uma porta ou janela seriam de algumas dezenas de pixels.

Os resultados mostram, porém, que o SSP é extremamente robusto ao ruído — indicando que mesmo uma segmentação grosseira permitiria obter bons resultados. De fato, os números obtidos pelo algoritmo com a perturbação de ruído são até mesmo ligeiramente melhores que o SSP sem ruído (mas a diferença não é significativa, *p-value* = 0,12) e mantêm uma diferença significativa para o método SPM, com *p-value* = 0,0008. A ligeira melhora tanto se explica pela flutuação aleatória como pelo fato de que o treinamento com as regiões perturbadas pode melhorar ligeiramente a capacidade de generalização do modelo.

O SSP é, porém, muito mais sensível à deleção de regiões. A perda de acurácia é significativa tanto no caso de deleções apenas quanto no caso de deleções com ruído, e o teste-t pareado com o SSP sem perturbações mostra *p-values* respectivos de 0,014 e 0,029. Nesses dois casos, a diferença entre o SSP e a SPM não é mais significativa (*p-values* respectivos de 0,13 e 0,15).

Esses resultados sugerem que o tratamento de regiões não detectadas é crítico para o bom funcionamento do SSP. Regiões importantes que não sejam detectadas (como os casos que citamos acima) deveriam ser tratadas como *missing-values*, necessitando portanto do tratamento estatístico adequado. Na implementação atual, se uma dessas regiões críticas desaparece (por exemplo, a única porta da fachada), o componente correspondente do vetor ficará vazio, perturbando consideravelmente a representação.

7.4 Análise dos Resultados

Para a classificação automática de imagens em classes, foi proposta uma nova forma de representação, baseada em dicionários visuais, que estende o conceito da divisão de imagens em níveis, do método SPM. A incorporação das regiões semânticas, pela técnica SSP, no processo de codificação das imagens, foi avaliada na separação entre imagens barrocas e modernas na cidade de Ouro Preto.

A classificação estilística de fachadas realizada na base de dados OP1K pelo método SSP superou a técnica SPM [Lazebnik et al., 2006], se mostrando adequada à classificação de fachadas barrocas e modernas. O ganho médio de acurácia da classificação do método SSP em relação ao SPM ficou em torno de 1,13 ponto percentual.

Tabela 7.1: Comparação das taxas de classificação para as técnicas SSP, SPM e BoW, e das as taxas alcançadas após a inserção de perturbações nas regiões semânticas usadas pela técnica SSP. Os intervalos de confiança são dados para $\alpha = 0{,}05$.

Valores de Referência	
Codificação	Acurácia
BOW	$92{,}80\% \pm 0{,}61\%$
SPM	$93{,}00\% \pm 0{,}78\%$
SSP	$94{,}13\% \pm 0{,}89\%$
SSP + Perturbação	Acurácia
Nenhuma	$94{,}13\% \pm 0{,}89\%$
Deleção 15%	$93{,}50\% \pm 0{,}95\%$
Ruído $\sigma = 40$	$94{,}38\% \pm 0{,}90\%$
Ambas	$93{,}40\% \pm 0{,}96\%$

Ganhos próximos dos extremos da escala (0 ou 100%) são consideráveis porque a escala percentual não é linear. Assim, a melhoria provida pelo SSP corresponde a passar, aproximadamente, de um erro a cada 14 imagens classificadas, para um erro a cada 17 imagens classificadas.

Além disso, foi testada a robustez da técnica proposta através da perturbação das regiões anotadas para as imagens. A robustez às perturbações das regiões abre espaço para que, no futuro, a descoberta das regiões seja feita de forma automática. O bom funcionamento do SSP, entretanto, exigirá um melhor tratamento das regiões porventura não detectadas, possivelmente através das técnicas de tratamento em aprendizado estatístico dos *missing-values*.

7.5 Considerações

Neste capítulo, foi apresentado o protocolo experimental usado na classificação de estilos arquitetônicos, aplicado em bases de imagens contendo fachadas de cidades históricas.

Antes dos experimentos de classificação estilística, uma breve caracterização do estilo barroco mineiro foi usada para justificar a escolha das regiões empregadas na codificação pela técnica SSP.

A partir disso, os resultados da classificação de estilos arquitetônicos pelas técnicas SSP e SPM [Lazebnik et al., 2006] foram apresentados e foi realizada, ainda, um

avaliação da robustez da codificação SSP proposta, através da introdução de perturbações nas regiões anotadas.

Finalmente, os resultados obtidos foram discutidos e desdobramentos futuros da pesquisa foram endereçados.

Capítulo 8

Conclusões

A busca e classificação em grandes bases de imagens, incluindo as do tipo *street-view*, demandam a criação de soluções automatizadas, já que o tratamento manual desses dados é inviável.

A solução passa pela extração de características, através dos descritores de imagens. As melhores soluções hoje disponíveis aliam o poder de discriminação dos chamados descritores locais, à compacidade e capacidade de generalização das representações globais, agregadas, no modelo que ficou conhecido como *bags of (visual) words*.

Neste trabalho, abordamos o problema de recuperação de uma cena específica em bases de *street-view*. A recuperação de uma cena, objeto ou imagem específicos é, usualmente, a tarefa em que o uso de descritores locais apresenta a maior efetividade. Entretanto, no caso das bases de *street-view*, a efetividade é impedida pela presença de uma grande quantidade de descritores espúrios, gerados por zonas altamente texturizadas, notadamente a vegetação e as sombras projetadas por esta, mas também elementos de fachada e calçamento na presença de iluminação rasante. Neste trabalho propusemos aliviar o problema pela filtragem desses descritores espúrios através de uma abordagem não-supervisionada.

Isso nos levou a estudar a clusterização nos espaços de características multimídia, que são desafiadores devido à alta dimensionalidade. Concentramo-nos nos métodos chamados de clusterização em subespaço, que são capazes de encontrar agrupamentos significativos em subconjuntos das dimensões, sendo, então resistentes ao problema da perda de contraste das distâncias no espaço de todas as dimensões. Propusemos uma técnica que alia a efetividade da família dos agrupamentos baseados em *mean-shift* à rapidez dos métodos de votação, como o *FINDIT*, transpondo a agressiva, porém cuidadosamente controlada subamostragem utilizada nos últimos aos primeiros. O algoritmo

Enhanced Mean-Shift for Subspace Clustering (E–MSSC) proposto se mostrou até 60 vezes mais rápido que o método não-otimizado, com perda moderada de precisão.

Essa perda se tornou tanto mais irrelevante, quanto notamos que para nossa aplicação específica, os resultados da identificação de cenas após a aplicação do método E–MSSC superaram aqueles obtidos com os algoritmos MSSC e FINDIT. Isso nos faz relembrar que a informação codificada pelos descritores já é, em si, intrinsecamente aproximada, e sujeita a um componente de erro aleatório não desprezível. Dessa forma, em se tratando de descritores multimídia, a insistência em técnicas exatas faz pouco sentido.

A técnica de separação de descritores, empregando aprendizagem não-supervisionada, seguida do casamento de imagens, se mostrou eficaz e ofereceu, nos experimentos feitos, uma melhora significativa na acurácia da identificação de cenas dentro de uma base de dados de cenas urbanas. Apesar disso, esbarramos numa limitação essencial da técnica não-supervisionada, que é a escolha dos clusters a considerar e daqueles a descartar: essa escolha exige, intrinsecamente, o exame dos resultados, e precisa ser feita por uma etapa posterior, supervisionada.

A segunda aplicação-chave abordada nesse trabalho diz respeito à classificação automática de imagens, especificamente no que toca à identificação de estilo arquitetônico em bases de *street-view*. Para essa aplicação, apresentamos uma extensão do modelo de *bags of visual words* que leva em conta regiões semanticamente relevantes, responsáveis por destacar objetos importantes na definição da classe à qual a imagem pertence. Essa nova técnica de representação de imagens, *Semantic Spatial Pyramids* (SSP), inspirada no método *Spatial Pyramids Matching* (SPM), foi aplicada em duas bases de imagens da cidade de Ouro Preto–MG, visando a identificação do chamado estilo "Barroco Mineiro", mostrando-se bastante promissora.

Apesar do sucesso da metodologia de classificação de imagens desenvolvida, ainda permanecem os desafios de se otimizar a descoberta e anotação das regiões semânticas, que podem ser feitas através de algoritmos de segmentação encontrados na literatura e aprendizagem supervisionada, além da aplicação da metodologia em bases de dados da literatura, das quais, algumas oferecem, inclusive, regiões delimitadas e anotadas para avaliação das técnicas de classificação.

8.1 Trabalhos Futuros

O estudo da etapa supervisionada da filtragem de descritores, especialmente no que toca à capacidade de generalização da escolha dos clusters de um conjunto de aprendizado

relativamente modesto para uma grande base de descritores é um alvo imediato da continuidade desse trabalho. A passagem ao universo supervisionado exige, entretanto, cuidados especiais, de forma a não haver contaminação indireta entre os conjuntos de treino e teste (pois imagens diferentes nas bases de street-view podem conter a mesma fachada, em parte, ou em todo). Essa foi uma das dificuldades, aliás, enfrentadas durante a execução dos experimentos da classificação estilística, mas aqui consideramos que tomar imagens de lados opostos da rua bastava para sanitizar as ameaças de contaminação, pois a presença de porções comuns relativamente modestas de objetos irrelevantes (carros, postes, partes do calçamento, objetos do fundo, etc.) não chega a ser um problema, já que estamos interessados na classificação das fachadas como um todo. Na tarefa de classificação dos descritores individuais, entretanto, parece-nos que os cuidados contra a contaminação cruzada devem ser mais estritos.

As duas técnicas propostas, o E-MSSC e as *Semantic Spatial Pyramids*, encontram aplicabilidade potencial em muitas outras tarefas além das imagens de street-view. Em trabalhos futuros gostaríamos de avaliar o E-MSSC em tarefas gerais de clusterização em alta dimensionalidade, para uma gama mais ampla de aplicações em bases reais e sintéticas, em que o MSSC ou o FINDIT se aplicam.

Gostaríamos também de avaliar as SSP em diferentes tipos de imagens em que seja possível encontrar diferentes regiões semanticamente relevantes claramente demarcadas. Acreditamos que na área de imagens médicas há uma grande oportunidade de contribuição, pois são imagens em que a aparência de zonas bastante específicas é de enorme importância para a interpretação e auxílio ao diagnóstico, e o grosseiro recorte das Pirâmides Espaciais comuns possivelmente não baste para codificar todas as nuances necessárias.

8.1.1 Cidades Históricas Mineiras

Futuramente, serão construídas bases de dados para cada uma das outras cidades históricas filmadas: Congonhas do Campos, São João del Rei e Tiradentes. As imagens de busca para essas bases serão geradas por um dispositivo móvel ou pelo uso de imagens das filmagens que não entrarem na base de cenas da respectiva cidade.

As novas bases servirão para validar, também, a aplicação da metodologia de representação de imagens por Pirâmides Espaciais Semânticas para identificação dos estilos arquitetônicos das construções.

Referências Bibliográficas

Aggarwal, C. C.; Wolf, J. L.; Yu, P. S.; Procopiuc, C. & Park, J. S. (1999). Fast algorithms for projected clustering. Em *ACM SIGMOD International Conference on Management of Data*, pp. 61--72.

Ali, H.; Seifert, C.; Jindal, N.; Paletta, L. & Paar, G. (2007). Window detection in facades. Em *Proceedings of the 14th International Conference on Image Analysis and Processing*, ICIAP '07, pp. 837 –842.

Ávila, A.; Gontijo, J. M. M. & Machado, R. (1996). *Barroco mineiro, glossário de arquitetura e ornamentação*. Coleção mineiriana. Fundação João Pinheiro.

Avila, S.; Thome, N.; Cord, M.; Valle, E. & de A. Araujo, A. (2011). Bossa: Extended bow formalism for image classification. Em *Proceedings of the 18th IEEE International Conference on Image Processing*, ICIP '11, pp. 2909--2912.

Bay, H.; Tuytelaars, T. & Gool, L. V. (2006). Surf: Speeded up robust features. Em *Proceedings of the 9th European Conference on Computer Vision*, ECCV '06, pp. 404--417.

Bellman, R.; Kalaba, R. & Zadeh, L. A. (1966). Abstraction and pattern classification. *Journal of Mathematical Analysis and Applications*, 13(1):1--7.

Bellman, R. E. (1957). *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edição.

Bimbo, A. D. (1999). *Visual information retrieval*. Morgan Kaufmann.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Boiman, O.; Shechtman, E. & Irani, M. (2008). In defense of nearest-neighbor based image classification. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '08, pp. 1--8.

Boureau, Y.; Bach, F.; LeCun, Y. & Ponce, J. (2010). Learning mid-level features for recognition. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '10.

Chang, C. & Lin, C. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1--27.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790--799.

Comaniciu, D. & Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603--619.

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273--297.

Csurka, G.; Dance, C. R.; Fan, L.; Willamowski, J. & Bray, C. (2004). Visual categorization with bags of keypoints. Em *Proceedings of the Workshop on Statistical Learning in Computer Vision*, ECCV '04, pp. 1--16.

da S. Torres, R. & Falcão, A. X. (2006). Content-based image retrieval: Theory and applications. *Theoretical and Applied Informatics (RITA)*, 13(2):161--185.

de M. Coelho, M.; Valle, E.; dos Santos Júnior, C. E. & de A. Araújo, A. (2011). Subspace clustering for information retrieval in urban scene databases. Em *Proceedings of the 24th Conference on Graphics, Patterns and Images*, SIBGRAPI '11, pp. 173--180.

de Mello, S. (1985). *Barroco Mineiro*. Coleção Primeiros vôos. Editora Brasiliense.

Doersch, C.; Singh, S.; Gupta, A.; Sivic, J. & Efros, A. A. (2012). What makes paris look like paris? *ACM Transactions on Graphics*, 31(4).

Duda, R. O.; Hart, P. E. & Stork, D. G. (2001). *Pattern Classification*. Wiley-Interscience, 2 edição.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32--57.

Everingham, M.; Gool, L. V.; Williams, C. K. I.; Winn, J. & Zisserman, A. (2007). The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

Fischler, M. A. & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications ACM*, 24:381--395.

Gan, G.; Ma, C. & Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics, Philadephia, PA.

Goil, S.; Nagesh, H. & Choudhary, A. (1999). Mafia: Efficient and scalable subspace clustering for very large data sets. Relatório técnico 9906-010, Northwestern University.

Griffin, G.; Holub, A. & Perona, P. (2007). Caltech-256 object category dataset. Relatório técnico 7694, California Institute of Technology.

Guha, S.; Rastogi, R. & Shim, K. (1998). Cure: an efficient clustering algorithm for large databases. Em *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 73--84.

Hathaway, R. J. & Bezdek, J. C. (2001). Fuzzy c-means clustering of incomplete data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 31(5):735--744.

Kläser, A.; Marszałek, M.; Laptev, I. & Schmid, C. (2010). Will person detection help bag-of-features action recognition? Relatório técnico RR-7373, INRIA Grenoble - Rhône-Alpes.

Kriegel, H.; Kröger, P. & Zimek, A. (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1--58.

Laptev, I.; Marszalek, M.; Schmid, C. & Rozenfeld, B. (2008). Learning realistic human actions from movies. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '08, pp. 1--8.

Laptev, I. & Perez, P. (2007). Retrieving actions in movies. Em *Proceedings of the 11th IEEE International Conference on Computer Vision*, ICCV '07, pp. 1--8.

Lazebnik, S.; Schmid, C. & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '06, pp. 2169--2178.

Li, F.; Fergus, R. & Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Em *Proceedings of the IEEE Workshop of Generative Model Based Vision*, WGMBV-CVPR '04.

Liu, C.; Yuen, J. & Torralba, A. (2009). Nonparametric scene parsing: Label transfer via dense scene alignment. Em *Proceedings of the IEEE Computer Vision and Pattern Recognition*, CVPR '09, pp. 1972--1979.

Liu, C.; Yuen, J.; Torralba, A.; Sivic, J. & Freeman, W. T. (2008). Sift flow: Dense correspondence across different scenes. Em *Proceedings of the 10th European Conference on Computer Vision: Part III*, ECCV '08, pp. 28--42.

Liu, C. L. (1968). *Introduction to combinatorial mathematics*. Computer science series. McGraw-Hill.

Lopes, A. P. B.; da S. Santos, E. R.; Valle, E.; de Almeida, J. M. & de A. Araújo, A. (2011). Transfer learning for human action recognition. Em *Proceedings of the 24th Conference on Graphics, Patterns and Images*, SIBGRAPI '11.

Lopes, A. P. B.; Oliveira, R. S.; de Almeida, J. M. & de A. Araújo, A. (2009). Spatio-temporal frames in a bag-of-visual-features approach for human actions recognition. Em *Proceedings of the 22th Conference on Graphics, Patterns and Images*, SIBGRAPI '09, pp. 315--321.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91--110.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. Em *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281--297.

Marszalek, M.; Laptev, I. & Schmid, C. (2009). Actions in context. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '09, pp. 2929--2936.

Martin, D.; Fowlkes, C. & Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530--549.

Mathias, M.; Martinovic, A.; Weissenberg, J.; Haegler, S. & Gool, L. V. (2011). Automatic architectural style recognition. Em *3D-ARCH 2011: 3D Virtual Reconstruction and Visualization of Complex Architecture*.

Mikolajczyk, K. & Schmid, C. (2001). Indexing based on scale invariant interest points. Em *Proceedings of the 8th International Conference on Computer Vision*, volume 1 of ICCV'01, pp. 525--531.

Nokia (2009). Nokia challenge (2009/2010): Where was this photo taken, and how?

Oliva, A. & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145--175.

Parsons, L.; Haque, E. & Liu, H. (2004). Subspace clustering for high dimensional data: a review. *SIGKDD Explorations Newsletter*, 6:90--105.

Patrikainen, A. & Meila, M. (2006). Comparing subspace clusterings. *IEEE Transactions on Knowledge and Data Engineering*, 18:902--916.

Penatti, O. A. B.; Valle, E. & da S. Torres, R. (2012). Comparative study of global color and texture descriptors for web image retrieval. *Journal of Visual Communication and Image Representation*, 23(2):359--380.

Perronnin, F.; Sánchez, J. & Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. Em *Proceedings of the IEEE European Conference on Computer Vision*, ECCV '10, pp. 143--156.

Philbin, J.; Chum, O.; Isard, M.; Sivic, J. & Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '07, pp. 1--8.

Picard, D.; Cord, M. & Valle, E. (2009). Study of sift descriptors for image matching based localization in urban street view context. Em *Proccedings of City Models, Roads and Traffic — ISPRS Workshop*, CMRT '09, pp. 193--198.

Recky, M. & Leberl, F. (2010a). Window detection in complex facades. Em *Proceedings of the 2nd European Workshop on Visual Information Processing*, EUVIP '10, pp. 220--225.

Recky, M. & Leberl, F. (2010b). Windows detection using k-means in cie-lab color space. Em *Proceedings of the 20th International Conference on Pattern Recognition*, ICPR '10, pp. 356--359.

Rose, K.; Gurewitz, E. & Fox, G. C. (1990). Statistical mechanics and phase transitions in clustering. *Physical Review Letters*, 65(8):945--948.

Ruspini, E. H. (1969). A new approach to clustering. *Information and Control*, 15:22--32.

Russell, B.; Efros, A. A.; Sivic, J.; Freeman, B. & Zisserman, A. (2009). Segmenting scenes by matching image composites. Em *Proceedings of Advances in Neural Information Processing Systems*, NIPS '09, pp. 1580--1588.

Russell, B.; Torralba, A.; Murphy, K. & Freeman, W. T. (2007). Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1–3):157--173.

Shalunts, G.; Haxhimusa, Y. & Sablatnig, R. (2011). Architectural style classification of building facade windows. Em *Advances in Visual Computing*, volume 6939 of Lecture Notes in Computer Science, pp. 280--289.

Shalunts, G.; Haxhimusa, Y. & Sablatnig, R. (2012a). Architectural style classification of domes. Em *Advances in Visual Computing*, volume 7432 of Lecture Notes in Computer Science, pp. 420--429.

Shalunts, G.; Haxhimusa, Y. & Sablatnig, R. (2012b). Classification of gothic and baroque architectural elements. Em *Proceedings of the 19th International Conference on Systems, Signals and Image Processing*, IWSSIP '12, pp. 316--319.

Sivic, J.; Russell, B. C.; Efros, A. A.; Zisserman, A. & Freeman, W. T. (2005). Discovering objects and their location in images. Em *Proceedings of the IEEE International Conference on Computer Vision*, number 4 in ICCV '05, pp. 370--377.

Sivic, J. & Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. Em *Proceedings of the 9th IEEE International Conference on Computer Vision*, ICCV '03, pp. 1470--1477.

Snavely, N.; Seitz, S. M. & Szeliski, R. (2006). Photo tourism: exploring photo collections in 3d. *ACM Transactions Graphics*, 25:835--846.

Torralba, A.; Fergus, R. & Weiss, Y. (2008). Small codes and large image databases for recognition. CVPR '08, pp. 1--8.

Turcot, P. & Lowe, D. (2009). Better matching with fewer features: The selection of useful features in large database recognition problems. Em *Proceedings of the IEEE*

12th International Conference on Computer Vision Workshops, ICCV Workshops '09, pp. 2109--2116.

Tuytelaars, T. & Mikolajczyk, K. (2008). Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*, 3:177--280.

Valle, E.; Cord, M. & Philipp-Foliguet, S. (2008). High-dimensional descriptor indexing for large multimedia databases. Em *Proceedings of the 17th ACM conference on Information and Knowledge Management*, CIKM '08, pp. 739--748.

Valle, E.; Picard, D. & Cord, M. (2009). Geometric consistency checking for local-descriptor based document retrieval. Em *Proceedings of the 9th ACM Symposium on Document Engineering*, DocEng '09, pp. 135--138.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Vogel, J. & Schiele, B. (2007). Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision*, 72:133--157.

Woo, K.; Lee, J.; Kim, M. & Lee, Y. (2004). Findit: a fast and intelligent subspace clustering algorithm using dimension voting. *Information and Software Technology*, 46(4):255--271.

Xu, R. & Wunsch, D. (2009). *Clustering*. Wiley-IEEE Press.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3):338--353.

Zhang, H.; Xiao, J. & Quan, L. (2010). Supervised label transfer for semantic segmentation of street scenes. Em *Proceedings of the 11th European Conference on Computer Vision: Part V*, ECCV '10, pp. 561--574.

Apêndice A

Aquisição das Bases de Dados

O contexto dessa aquisição é o Projeto Cidade Virtual[1] que visa a construção de um repositório de imagens de cidades tombadas como Patrimônio Histórico. O objetivo do projeto é realizar uma documentação visual dessas cidades pela captura de imagens *street–view* e, então, oferecer uma interface de navegação e um passeio virtual[2]. Nesta seção é discutido o sistema empregado na aquisição das imagens *street–view*.

A.1 Sistema de Aquisição

Inicialmente, idealizou-se o uso de um conjunto de câmeras integradas e que fossem sincronizadas e disparadas automaticamente, uma vez que isso resultaria em uma aquisição de imagens muito menos trabalhosa. Assim, um sistema composto de seis câmeras foi adquirido, a Ladybug 2[3], com cada câmera capaz de capturar imagens com resolução de 1024×768 pixels a uma taxa de 30 quadros por segundo. Além disso, o sistema promove a compactação em tempo real do fluxo de vídeo gerado via hardware, integra esse fluxo de vídeo com coordenadas obtidas por um receptor *Global Positioning System* (GPS) e armazena todos os dados em um único arquivo de vídeo.

Posteriormente, um sistema veicular foi projetado para prover suprimento de energia, localização através de receptor GPS, integração com sistema computacional e armazenamento adequado para os dados gerados pelas câmeras, conforme pode ser visto na Figura A.1. Foi usada a seguinte relação de equipamentos:

1. Uma bateria selada de 12V e com capacidade de fornecimento de 40mAh;

[1] www.npdi.dcc.ufmg.br/projects/CidadeVirtual
[2] www.npdi.dcc.ufmg.br/projects/viewer
[3] http://www.ptgrey.com/products/ladybug2/

2. Um inversor de tensão 12V/110V para alimentação dos equipamentos;
3. Um notebook Sony Vaio FZ485U/B equipado com processador Core 2 Duo T8100, 4GB de memória RAM, placa de vídeo NVIDIA GeForce 8400GT e um *PCI–Express Card* com interface IEEE–1394B para ligação com a câmera Ladybug 2;
4. Um disco externo de 1,5TB para armazenamento dos dados;
5. Um receptor de sinal GPS com interface Bluetooth;
6. Uma câmera Ladybug 2 com resolução de 1024 x 768 pixels e taxa de captura de 30 quadros por segundo.

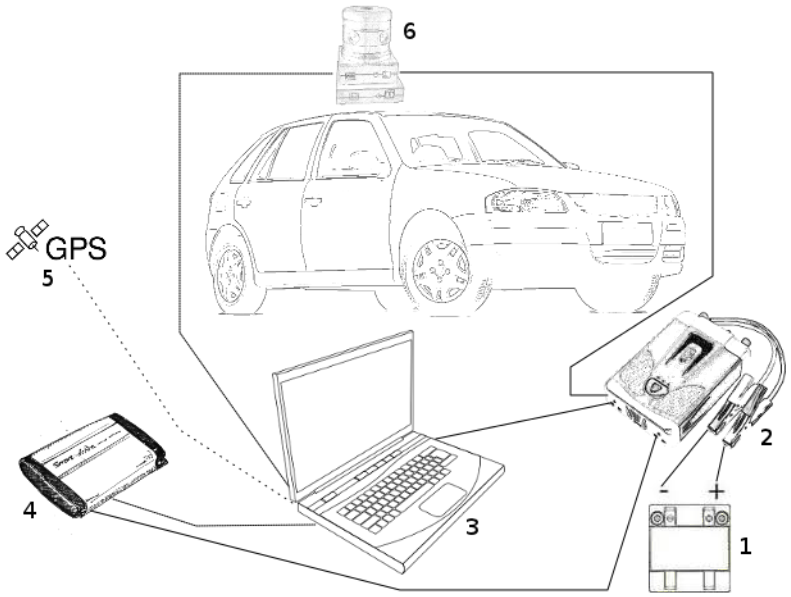


Figura A.1: Esquema geral de funcionamento dos equipamentos de filmagem: (1) Bateria, (2) Inversor de Tensão, (3) Notebook, (4) Disco Externo, (5) Receptor GPS, (6) Ladybug 2

Em um primeiro momento, a ideia era conectar ao computador uma unidade de armazenamento externo com capacidade de 1TB, alimentada pelo inversor e usando uma interface USB 2.0. Mais tarde, optou-se por customizar uma unidade de armazenamento, montando um disco convencional de 7200 RPM e 1,5TB em um estojo apropriado. Esse estojo é feito de alumínio e possui exautores ativos, prevenindo, assim, problemas de superaquecimento.

Por fim, um GPS Bluetooth foi usado para aquisição de coordenadas geográficas como entrada para o software da câmera. Uma taxa de atualização de 5Hz no GPS foi

suficiente para atribuir marcação geográfica dos quadros do vídeo, ainda mais se for levada em consideração a baixa velocidade do veículo.

A fonte de energia seria, inicialmente, a própria bateria do veículo usado na captura das imagens, conectada a um Inversor de Energia 12Vcc/115Vca, com o objetivo de simplificar o sistema. Entretanto, nos primeiros testes, ficou claro que a bateria do veículo não teria potência suficiente para alimentar tantos dispositivos. Sendo assim, o projeto foi modificado para empregar baterias avulsas, capazes de fornecer até 12Vcc, 40mAh e aproximadamente quatro horas de aquisição contínua de vídeo.

Na montagem do sistema de captura, a parte mais desafiadora foi a colocação da câmera sobre o teto do veículo, devido a questões de vibração, estabilização da imagem e distribuição de peso. A solução adotada foi empregar um *rack* veicular padrão sobre o carro e construir um suporte de câmera para ser fixado nele e receber a Ladybug 2.

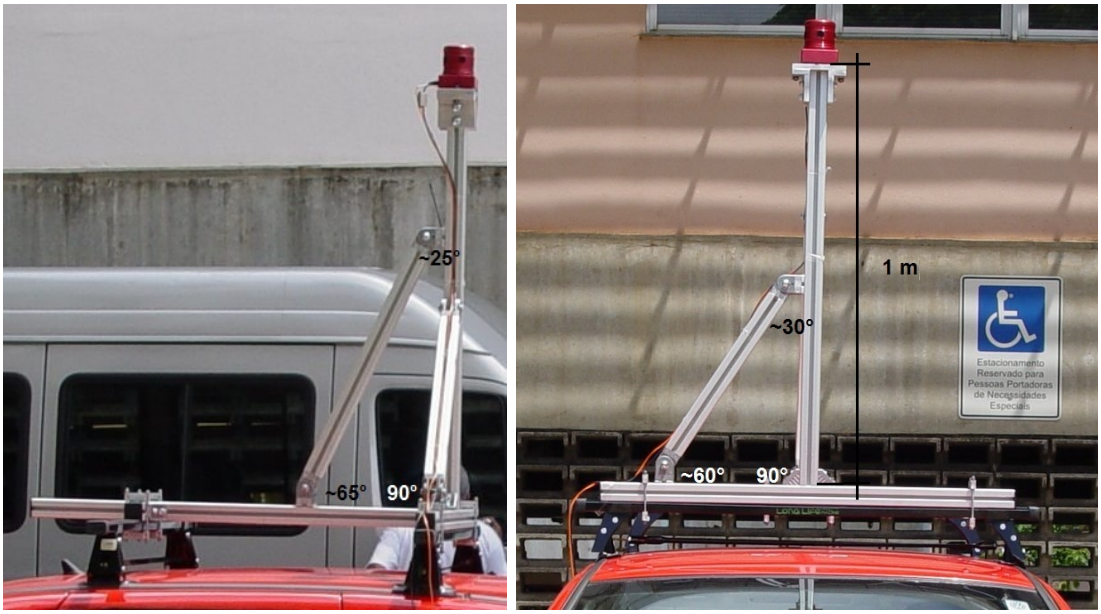
Na Figura A.2, o suporte da câmera e sua estrutura são mostrados. O suporte foi construído utilizando perfis de alumínio Bosch Rexroth[4], escolhidos por serem extremamente leves e resistentes, evitando, assim, problemas com a distribuição de peso. O projeto foi concebido para minimizar a oscilação da câmera e é composto de duas barras oblíquas para inibir movimentos laterais e longitudinais. A barra estabilizadora longitudinal, com 70 cm de comprimento, é mostrada na Figura A.2a. A barra lateral, com cerca de 50 cm de comprimento, faz um ângulo de 30° com a barra que suporta a câmera e um ângulo de 60° com base do suporte (Figura A.2b).

Esse aparato foi testado em campo com sucesso, sendo as vibrações longitudinais e laterais minimizadas. As condições de teste foram particularmente desafiadoras, porque as cidades que são Patrimônios Culturais possuem muitas ruas calçadas com blocos de pedra bem irregulares. Além disso, havia uma limitação de velocidade a ser obedecida para a aquisição das imagens (ver Seção A.2)

A câmera foi colocada a, aproximadamente, um metro acima do teto do veículo, numa tentativa de manter partes do mesmo fora do campo de visão. Esta distância foi obtida empiricamente, com o uso de um tripé sobre uma superfície plana de tamanho semelhante ao do teto do veículo. Diversas alturas foram experimentadas até que a superfície não aparecesse no vídeo.

Na preparação final para as filmagens, todos os conjuntos de parafusos e porcas foram combinados com arruelas de pressão e apertados firmemente para evitar que os parafusos se soltassem e, consequentemente, que o suporte da câmera caísse.

[4]http://www.boschrexroth.com/business_units/brl/en/produkte/mge/index.jsp



(a) (b)

Figura A.2: Montagem final do equipamento sobre o veículo: (a) Visão Lateral e (b) Visão Frontal

A.2 Captura dos Vídeos

Claramente, não se esperava que o suporte criado para carregar a câmera fosse capaz de eliminar completamente as oscilações, da mesma forma que um estabilizador devidamente projetado para esse fim o faria. Portanto, a ferramenta de estabilização de imagem, disponível no software que acompanha a Ladybug 2, permaneceu ativada durante as filmagens. Entretanto, não se evidenciou oscilações no vídeo final, mesmo sem usar a ferramenta de estabilização por software.

Para este trabalho, foi selecionada a cidade histórica mineira de Ouro Preto, tombada como Patrimônio Histórico da Humanidade pela Unesco, para descrição do processo de aquisição e teste das metodologias de casamento de imagens e classificação de estilos arquitetônicos. A estratégia de coleta das imagens foi dirigir pelo centro histórico e também pela periferia da cidade, com presença de novos bairros (com o propósito de permitir a comparação dos estilos arquitetônicos). Durante o trajeto, o veículo se moveu a cerca de 30 km/h enquanto que a interface da câmera indicava uma taxa de captura de 6 quadros por segundo, o que significa que uma imagem foi capturada a cada 1,39 m, aproximadamente. Para cobrir os cerca de 51 km de ruas (ocorrendo, inevitavelmente, redundâncias não intencionais devido à topologia da

Tabela A.1: Dados da filmagem de teste realizada na UFMG

Data	Local	Volume	Quadros	Tempo(min)	Distância(km)
10/2009	UFMG	43,8 GB	23.151	64	12,5

cidade), foram gastas 26 horas de filmagem e produzidos 331 GB de dados de vídeo, com compressão por hardware (ver Tabela A.2).

Ressalta-se que a compressão por hardware do fluxo do vídeo preserva a qualidade dos quadros, reduzindo o tamanho final de armazenamento dos dados. Mais tarde, tanto imagens estáticas (usadas neste trabalho) como fluxos de vídeo com um grau maior de compressão (usados em outras aplicações, como a interface de navegação) foram extraídos a partir do fluxo de vídeo original de alta qualidade.

No desenrolar das filmagens, a maior dificuldade foi a falta de sinal GPS na região de Ouro Preto, na data planejada para geração do vídeo. A situação foi verificada em um outro dispositivo usado pela equipe, descartando falha do equipamento. A alocação de um período maior para a realização das filmagens poderia resolver o problema, tornando possível esperar por condições meteorológicas favoráveis. Ao contrário do que se esperava, o consumo de espaço nas unidades de armazenamento foi inferior a 1 TB, evitando interrupções para substituir discos externos, o que pode ser atribuído à compressão por hardware citada anteriormente.

No mês de novembro de 2009 foi realizada uma filmagem para testar o equipamento no Campus Pampulha da UFMG cujos dados podem ser vistos na Tabela A.1 e o percurso feito na Figura A.3.

Feito isso, foi executada a filmagem na cidade de Ouro Preto–MG, nos dias 22 e 23 de março de 2010. O fluxo de vídeo foi capturado no formato proprietário da câmera, sem compressão adicional (além daquela feita por hardware). Apesar dos problemas de sinal encontrados, dados de georreferenciamento foram inseridos manualmente em momento posterior, permitindo a criação de mapas de trajeto. O volume de dados gerado pela captura do fluxo de vídeo é de 2GB/min, tendo sido totalizado em torno de 331,1GB de dados (Tabela A.2 e Figura A.4).

Numa etapa de pós-processamento, foram selecionados trechos do vídeo, com o objetivo de diminuir redundância, e as imagens foram anotadas manualmente, para que fossem identificadas as ruas às quais pertencem e permitir a avaliação automática dos resultados obtidos.

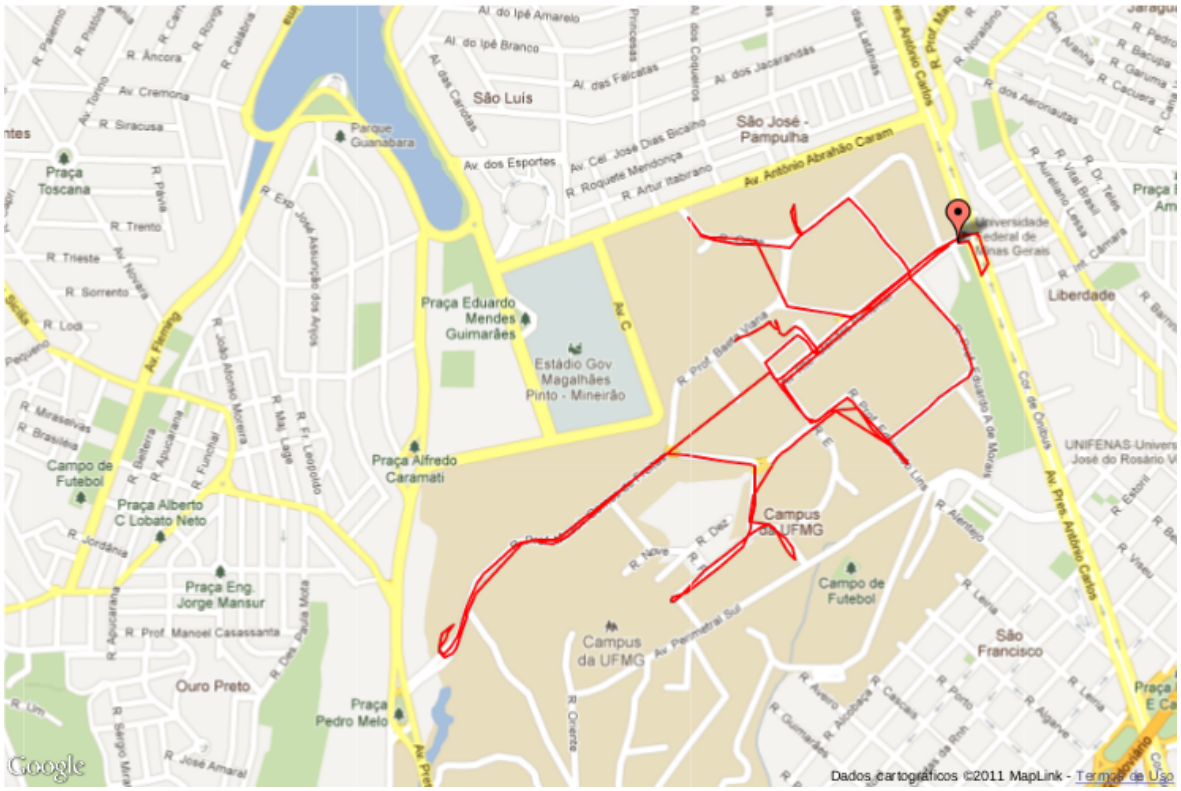


Figura A.3: Percurso da filmagem para teste do equipamento realizada na UFMG

Tabela A.2: Dados das filmagens realizadas para criação da base de cenas urbanas de cidades históricas mineiras

Data	Local	Volume (GB)	Quadros Filmados	Tempo de Filmagem (min)	Distância Percorrida (km)
22/03/2010	Ouro Preto	331,1	95.357	221	[5]51,0
08/09/2010	Congonhas do Campo	67,8	23.459	39	9,3
09/09/2010	São João del Rei	205,8	46.459	130	42,5
10/09/2010	Tiradentes	57,3	13.028	36	10,2

Tendo sido definido esse protocolo de criação das bases, mais três cidades foram filmadas, todas em Minas Gerais: Congonhas do Campo, São João del Rei e Tiradentes (Tabela[5] A.2 e Figuras A.5, A.6 e A.7).

A grande vantagem do procedimento de captura é que ele é totalmente portável, por não depender do meio de locomoção. Isso significa que ruelas não acessíveis por carro podem ser visitadas por uma bicicleta, rebocando um carrinho que contenha o

[5]Em decorrência da falta de sinal do GPS na região durante as filmagens, o valor para a distância percorrida em Ouro Preto é estimado.

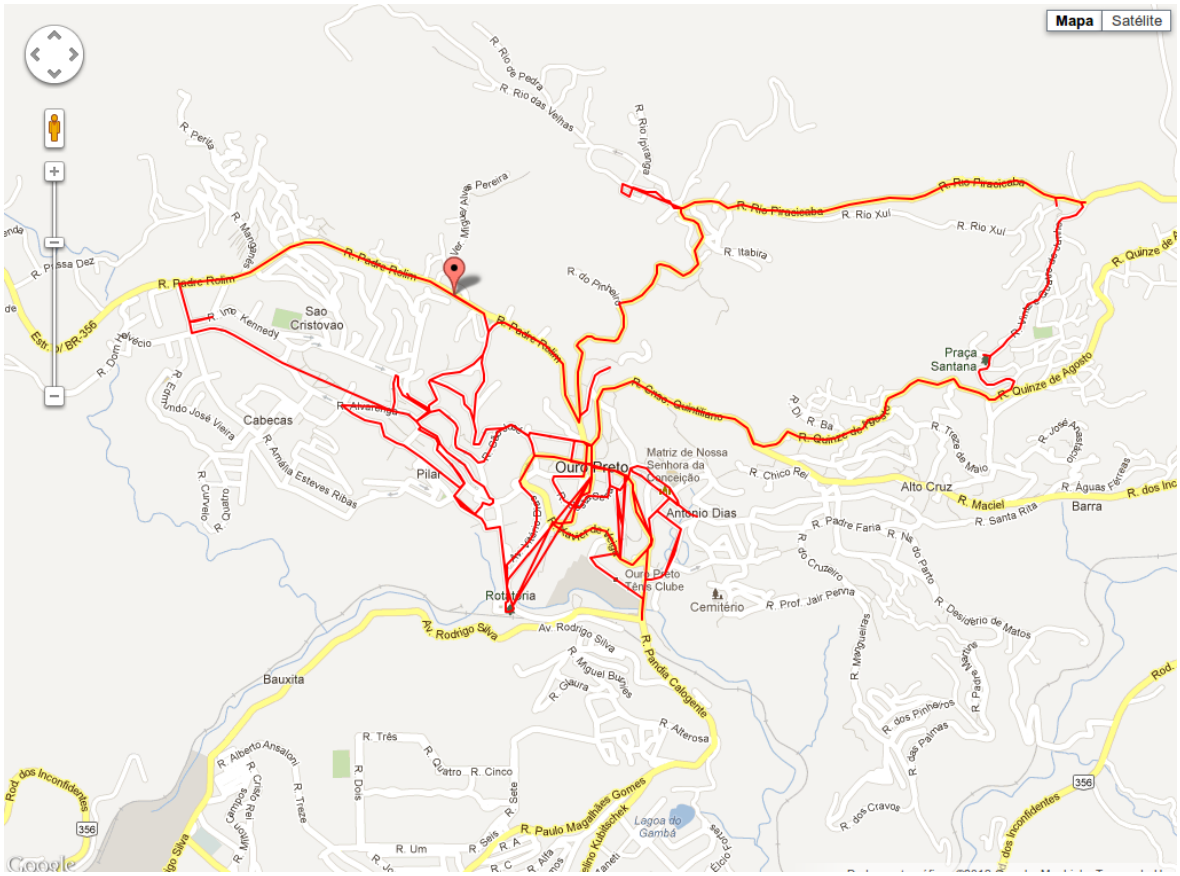


Figura A.4: Percurso de um dos trechos da filmagem realizada em Ouro Preto

equipamento e, ainda, interiores de igrejas e museus, por exemplo, podem ser imageados usando-se um pequeno transporte, empurrado pelo operador do equipamento.

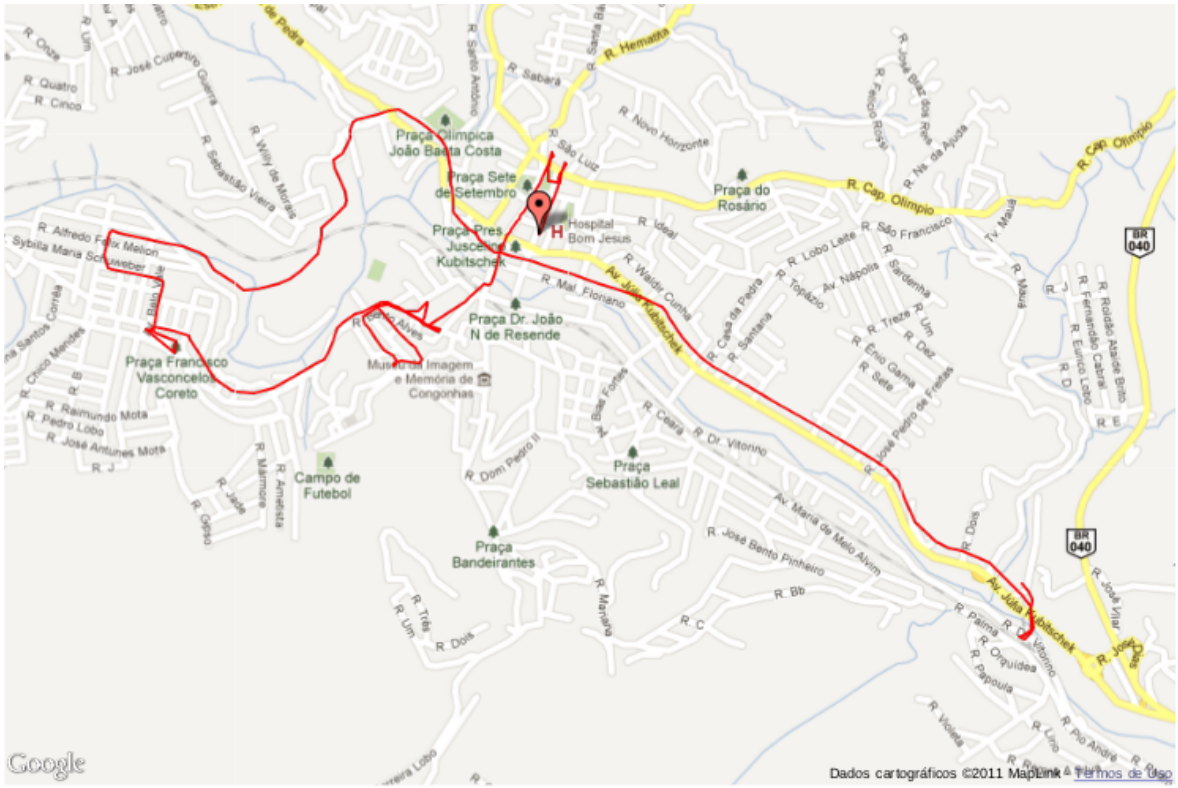


Figura A.5: Percurso da filmagem realizada em Congonhas do Campo

Figura A.6: Percurso de um dos trechos da filmagem realizada em São João del Rei

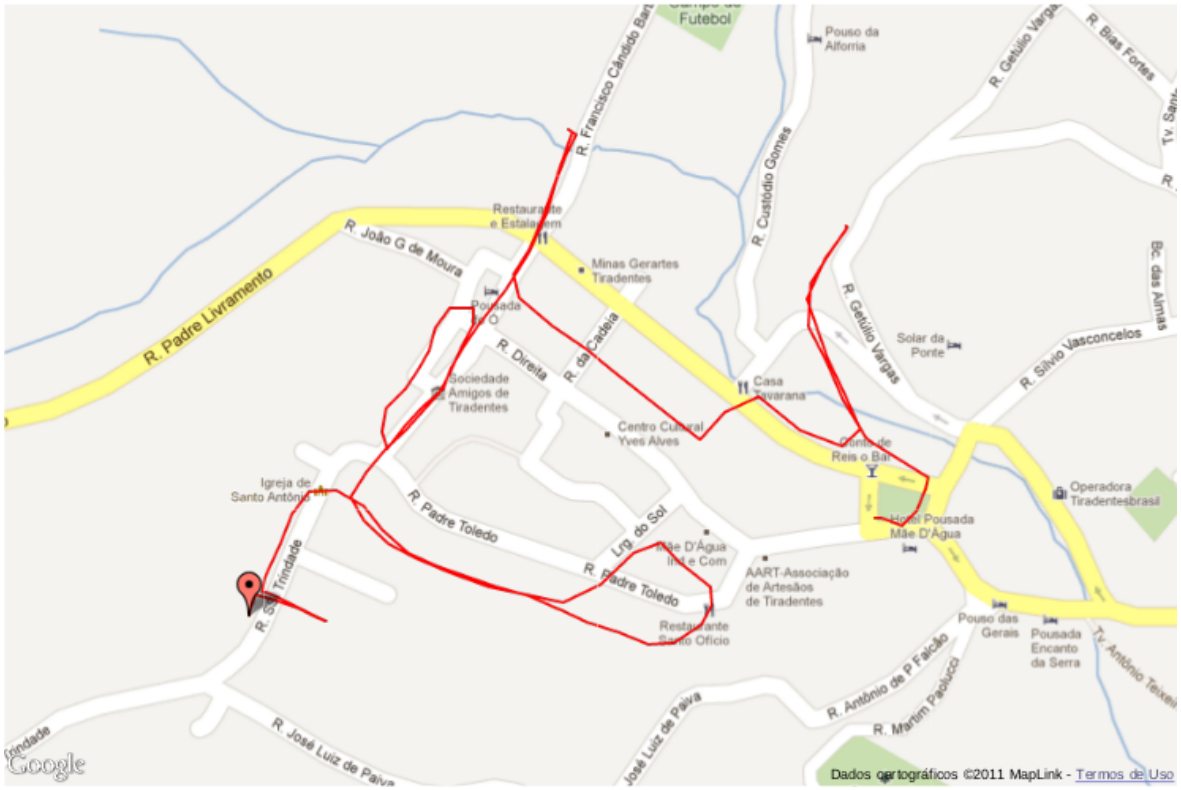


Figura A.7: Percurso de um dos trechos da filmagem realizada em Tiradentes

Apêndice B

Atividades desempenhadas durante o Doutorado

O objetivo deste capítulo é relacionar as principais atividades desempenhadas durante os estudos de doutoramento mas que não estiveram diretamente relacionadas com a tese, como coorientações, apresentação de trabalhos e divulgação na imprensa.

B.1 Coorientações

Para o desenvolvimento deste trabalho, diversos alunos de Iniciação Científica foram co-supervisionados pelo doutorando.

- Anderson N. A. Peixoto — Bolsista de Iniciação Científica do Projeto Cidade Virtual. Foi o primeiro bolsista do projeto, quando ainda não havia sido adquirido o equipamento para as filmagens na cidade de Ouro Preto. Sendo assim, o bolsista foi colaborador no projeto de detecção de nudez em imagens, ajudando na implementação do código em Matlab para a realização dos experimentos.

- Glauber Martins — Bolsista de Iniciação Científica do Projeto Cidade Virtual. Além de ajudar nos primeiros testes de captura dos vídeos, realizou as primeiras explorações na interface proprietária da câmera Ladybug 2, estendendo as APIs fornecidas pelo fabricante para limitar o campo de visão vertical das imagens e atuou também no estudo de algoritmos de fusão das imagens das câmeras.

- Cássio E. dos Santos Júnior — Bolsista de Iniciação Científica do Projeto Descoberta Semântica em Vídeos de Cidades Históricas Mineiras — Edital MCT/CNPq n.º 12/2010. Atuou em diversas frentes dentro do NPDI, tendo apoiado a pesquisa

sobre os algoritmos de clusterização, sendo o responsável pela implementação do algoritmo FINDIT a partir do estudo do artigo que apresentou o algoritmo, e contribuiu na portabilidade do código de amostragem da base de dados do FINDIT para o algoritmo MSSC.

- Eduardo Gomes Filho — Bolsista de Apoio Técnico — Edital MCT/CNPq 10/2010. Cooperou no gerenciamento do NPDI, tendo ajudado na configuração das máquinas para autenticação dos usuários na rede do DCC, quando da saída do NPDI do controle do Centro de Recursos Computacionais do DCC. Responsável, também, pela manutenção das máquinas do laboratório.

- Guilherme da Silva Nascimento — Bolsista de Apoio Técnico — Edital MCT/CNPq 10/2010. Substituiu o bolsista Eduardo na manutenção das máquinas do NPDI.

- Guilherme Leite — Coorientação de Projeto Orientado em Computação I e II. Desenvolveu uma interface de comunicação entre dispositivos móveis e desktops para captura de uma imagem pelo primeiro que seria enviada ao segundo para processamento do algoritmo de identificação de cenas, cujas respostas retornariam ao dispositivo inicial para serem apresentadas ao usuário.

B.2 Apresentação de Trabalhos

Dada a afinidade do Projeto Cidade Virtual com a Escola de Belas Artes da UFMG, houve o convite para que o mesmo fosse apresentado em um seminário institucional. O projeto também figurou na apresentação final do Projeto Orientado em Computação do aluno orientado Guilherme Leite.

- Apresentação da palestra intitulada "Projeto Panorâmica Virtual para Cidades Históricas — Cidade Virtual", nos Seminários de Pesquisa em Pós-Graduação (*Seminar of graduate research*). Realização: ARCHE e LACICOR. PPGA–EBA–UFMG — Belo Horizonte–MG, 02 de setembro de 2011.

- Apresentação, pelo estudante Guilherme Leite, do Projeto Orientado em Computação intitulado "Sistema de Recuperação de Informação Através de Casamento de Imagens". Orientado pelo Prof Arnaldo de Albuquerque de Araújo e coorientado por Marcelo de Miranda Coelho. DCC/UFMG, Belo Horizonte–MG, segundo semestre de 2010.

B.3 Difusão na Mídia

Com a visibilidade do Projeto Cidade Virtual, este trabalho encontrou interesse pela mídia destinada ao público geral, em duas matérias impressas e uma entrevista de rádio.

- Reportagem de capa no Boletim da UFMG, Nº 1.737 — Ano 37, de 09 de maio de 2011,p. 5.[1]
- Reportagem de meia página no Jornal Hoje em Dia, Edição 8.206, de 13/05/2011, Caderno Minas, p. 24.[2]
- Entrevista de 8 minutos à rádio UFMG Educativa no dia 16 de maio de 2011 para o programa Conexões.[3]

[1] http://www.ufmg.br/boletim/bol1737/5.shtml
[2] http://www.hojeemdia.com.br/minas/tour-virtual-em-cidade-historica-de-minas-1.279006
[3] http://www.ufmg.br/online/radio/arquivos/anexos/MARCELO%20COELHO%20-%20PROJETO%20CIDADE%20VIRTUAL%20-%2016-05-2011.mp3