THE ROLE OF TEXTURE IN INDOOR SCENE RECOGNITION

SHAHZOR AHMAD

(M.Sc. (Signal Processing), NTU)

A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

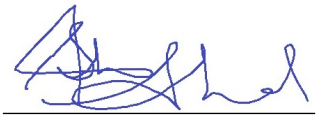
DEPARTMENT OF
ELECTRICAL & COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2016

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.



Shahzor Ahmad

Monday 4th July, 2016

Indeed, in the creation of the heavens and the earth and the alternation of the night and the day are signs for those endowed with intellect.

Al Qur'an 3:190

Acknowledgements

It has been a pleasure to have had Prof. Cheong Loong Fah as my doctoral advisor at NUS ECE, for whom I have the utmost respect. He was instrumental in inculcating a reading habit in me early on ("You must read, read, read at this stage!", he would say). This thesis has only been arrived at due to his expert guidance over the years, through countless insightful discussions with him, his pointers to pertinent literature (he had suggested I investigate TILT for scene layout as early as four years ago!), and his meticulous reviews of my writings. I am also thankful to him for his support with the many administrative issues with ECE and FOE.

Thank you, ECE, NUS and MOE for awarding me the NUS Research Scholarship to study for a Ph.D. in S'Pore — it's been one amazing experience! Thanks to Prof. Yan Shuicheng (whose group seminars I'd attend in my 2nd year, and who allowed me use of their LV cluster when I was exploring part discovery), my early mentor Dr. Gerhard Roth (with whom I'd have lengthy, exhilarating Friday afternoon discussions — Sivic's "Video Google" being one of them), Prof. Ashraf Kassim, Dr. Henry Duh Been-Lirn and Dr. Robby Tan.

I am grateful to ECE and FOE for giving me the opportunity to serve as a TA. Not only did that support me through my fifth year of study, it allowed me to teach — something I greatly enjoy, provided me with an everyday sense of accomplishment (not easily or frequently achieved for a Ph.D. student!), and helped me focus much better on my research. Thanks to my co-workers Dr. Rajesh Panicker, Dr. Chua Dingjuan, and particularly Christopher Moy Shin Lee Lan Chong, who were always considerate of the fact that I was also a research student.

It was a blessing having my brother Zohair Ahmad and long-time friend Arsalan Mansoor around, who facilitated a homely atmosphere. My special, heart-felt thanks to Muhammad Qasim Mehmood (occasional prayer and lunch buddy), Mansoor Shaukat (early library buddy) and Ahmed Mahmood (for his positivity), who were around for help and encouragement, and to fellow annotators Samuel Lee Zhi Wei and Pyae Phyo Tun, and so many other well-wishers.

It was a pleasure befriending lab mates Peilin Wang and Shintaro Kitazawa (who were also occasional buddies to the movies), the encouraging Choon Meng Lee and Luo Ye, the helpful, humble and friendly Zhou Qiang, Tran Lam An, Li Zhuwen, Guo Jiaming, Kou Wen, and Kaimo Lin, and especially Huang Rui.

Attending the weekly *halaqah* on Qur'anic exegesis at Al Qudwah Academy by Ustadh Hidayat Radja Nurul Bahri, and organized by Rizal Bro. among others, was a most uplifting, spiritual, yet scholarly experience during a challenging phase. I am indebted to Ustadh Ilyas Jamali whose *makhraj* classes I attended at Darul Arqam S'Pore, and thankful to my swim coach Russell Wang, whose training sessions were always a welcome respite from the week-long mental exertion.

Salam to the Muslim community in S'Pore, due to whom I have enjoyed *halal* cuisine, Fridays have been *Jumu'as*, the fasting months have felt like *Ramadhans*, and Hari Rayas like *Eids*!

In the end, I would like to thank my beloved family for their support, encouragement, prayers and patience during my long years in S'Pore.

To my parents

Contents

Abstract

Indoor scene recognition is the problem of assigning a semantic category to a given image depicting some indoor scene. A fundamental problem in computer vision, it has the potential to facilitate a holistic understanding of the scene, and thus favorably influence other tasks such as contextual reasoning and path planning in intelligent machines.

This thesis advances a novel paradigm involving the use of planar homogeneous texture for an improved scene representation and subsequent classification. Such texture manifests in the form of regularly repeating structural or architectural elements, or as uniform printed or engraved patterns on material, and is abundantly present in indoor scenes.

In order to mitigate in-class variation arising out of viewpoint differences and perspective projection in images, the problem of planar rectification of homogeneous texture is first addressed. A texture frequency projection model is developed in order to recover plane projective parameters, allowing an affine-ambiguous rectification. An existing scheme to recover dominant instantaneous frequency is examined in depth, identifying and successfully addressing its short-comings — frequency drift and quadrant ambiguity — via energy minimization methods. Robust parameter recovery is demonstrated, and a non-isotropic multi-scale representation proposed for improved estimates. Comprehensive qualitative and quantitative evaluations are presented, and the proposed scheme is shown to outperform existing representative work on texture rectification in real-world images marred with outliers, clutter and photometric severities.

Current approaches to detecting mid-level features use learning to automatically discover discriminative scene parts. This is essentially a chicken-and-egg problem, where neither part appearance models nor part instances in images are known. This thesis instead advocates and demonstrates the

detection of homogeneous texture in multi-planar, cluttered scenes via the texture projection model developed earlier, making for a hand-crafted approach to detect semantically meaningful mid-level features. At the same time, the detection is inherently projective-invariant (therefore, subsuming affine invariance), as opposed to existing low-level scale and rotation, or affine invariant blob and edge detectors. The proposed detection framework is qualitatively and quantitatively evaluated and shown to significantly outperform existing representative work.

Homogeneous texture as detected by the proposed method is shown to perform favorably in providing a crude geometric indoor layout in multi-planar textured scenes. In doing so, the approach sidesteps the error-prone, ill-posed computation of vanishing points in order to establish room orientation, and does not need to rely upon the simplistic Manhattan or box layout assumption, or to employ machine learning to localize room faces in space and scale, as does existing work.

Affine rectification of detected homogeneous texture is found to yield low-level features that are not only class-discriminative, but also complementary to regular, non-rectified features, thereby facilitating indoor scene recognition. The results are consistent across a number of hand-crafted descriptors, both thresholding (CENTRIST, LBP) and gradient based (SIFT, HOG), as well as pre-learned deep ConvNet features. Classification performance based on a combined feature representation is seen to favorably compare with contemporary approaches on the 67-category MIT Indoor benchmark spanning 6700 images, while one of the presented configurations outperforms most current state-of-the-art work. The proposed approach is additionally evaluated on a set of 31 categories spanning 6200 images (mostly outdoor, man-made environments exhibiting regular, repeating structure), being a subset of the Places2 large scale scene dataset.

List of Figures

List of Tables

List of Abbreviations (in Alphabetical Order)

ANN	Artificial Neural Network
BOW	Bag of Words
CENTRIST	CENsus TRansform hISTogram
DFT	Discrete Fourier Transform
CNN	Convolutional Neural Network
ConvNet	Convolutional Neural Network
DPM	Deformable Part Model
EM	Expectation Maximization
GCO	Graph Cut Optimization
GMM	Gaussian Mixture Model
HIK	Histogram Intersection Kernel
HOG	Histogram of Oriented Gradients
IFV	Improved Fisher Vecotr
KNN	K Nearest Neighbours
LBP	Local Binary Patterns
LLC	Locality constrained Linear Coding
MAP	Maximum A-Posteriori Estimation
MLE	Maximum Likelihood Estimation
MLP	Multi Layer Perceptron

MSER	Maximally Stable Extremal Regions
NMS	Non-Max Suppression
PCA	Principal Component Analysis
PMK	Pyramid Match Kernel
QPBO	Quadratic Pseudo Boolean Optimization
RANSAC	RANdom SAmple Consensus
RBF	Radial Basis Function
REM	Repetition Maximization
RMSE	Root Mean Squared Error
SFT	Shape From Texture
SIFT	Scale-Invariant Feature Transform
SP	Spatial Pyramid
SPM	Spatial Pyramid Matching
SR	Sparse Representation
STFT	Short-Term Fourier Transform
SVD	Singular Value Decomposition
SVM	Support Vector Machine
texels	texture elements
TILT	Transform Invariant Low-rank Texture

Chapter 1

Introduction

1.1 The Problem of Indoor Scene Recognition

Andrew Fitzgibbon, announcing the conferral of the 2008 British Machine Vision Association (BMVA) Distinguished Fellowship upon Andrew Zisserman — who had contributed significantly to multiple view geometry in computer vision — writes [37]:

> "Geometry was successful in showing that computer vision could solve problems which humans could not: recovering 3D structure from multiple images required highly trained photogrammetrists and took a considerable amount of time. However, Andrew's interests turned to a problem where a six-year old child could easily beat the algorithms of the day: object recognition."

Indeed, attaining human-level performance in visual recognition is a holy grail for computer vision. Where humans have an uncanny knack of recognizing objects – albeit any changes in size and appearance, or environmental conditions such as lighting – we are far from mimicking the same level of performance in machines.

As with objects, recognizing scenes comes to us humans naturally. The problem consists in assigning a semantic category to a given scene — e.g., a grassy flatland, open sky and sunlight are characteristic of a field, while the presence of furnishings such as sofas, chairs and rugs suggests the scene depicts a living room. Such a semantic categorization can potentially facilitate a holistic understanding of the scene, and favorably influence other research problems in computer vision such as contextual reasoning. It holds the key to building intelligent machines that can perform high-level tasks such as path planning and sensing obstacles, or to equip them with the ability to move and manipulate objects. This problem of recognizing *semantically similar* scenes is not to be confused with scene retrieval, also called place recognition, wherein the *physically same* scene or environment may be recognized from any of its given viewpoints. Retrieval is not the focus of our discussion.

Understandably the problem of semantic scene recognition is far more challenging than that of object recognition. An appropriate scene representation must be devised that can effectively capture the typicality of a certain scene category. Moreover if, for instance, one were to describe a scene in terms of the contained objects or regions (in order to compare it with the typical or exemplar representation one has 'learnt' from experiencing this category previously), a bottom-up appearance based 'segmentation' of a given scene into such parts is yet another, under-constrained, problem in

computer vision, though solved effortlessly by humans! Alternatively, characteristic parts of a sene may be 'detected' in a top-down fashion, and this comes with its own set of challenges — *what* parts should one detect that would consequently help in distinguishing a scene, and how does one establish the typical appearance characteristics of such parts? Additionally, any two photos of a given scene category, though semantically similar, can differ considerably in terms of contained objects (and, in turn, their appearances), viewpoints and lighting conditions, thereby making the task rather difficult to mimic in machines.

1.1.1 Problem Statement

The focus of this thesis is to identify and address some of the problems faced in performing *indoor* semantic scene recognition from the technical standpoint. We seek to obtain an improved scene representation, that can more efficiently encode the similarities among images of the same scene category. In this regard, the abundant presence of characteristic repeating patterns — called 'homogeneous texture' — in indoor scenes will be highlighted, and a robust pipeline that can effectively make use of such patterns devised. The role of such texture, which manifests either as printed/material or structural patterns, in providing a crude geometric layout in real-world indoor scenes, as well as recognition will be explored.

1.2 Summary of Contributions

This thesis makes the following important contributions:

- A novel paradigm advocating the use of characteristic repeating patterns, called homogeneous texture, for indoor scene recognition is motivated, as opposed to traditional learning based low-level or mid-level features.

- Prior work on planar projective rectification, particularly texture rectification is reviewed at length, its short-comings on real-world images exhibiting clutter, outliers and photometric severities are highlighted, and a frequency based approach is advocated to address these challenges. A novel texture frequency projection model is developed. An existing scheme to recover dominant instantaneous texture frequency is examined in depth, identifying and successfully addressing two short-comings — frequency drift and quadrant ambiguity — in real world images. Comprehensive qualitative and quantitative evaluations are presented, and the proposed scheme is shown to have a superior performance compared to existing representative work on texture rectification.

- The proposed projective rectification model is put to use for localizing potentially large homogeneous texture in real-world, cluttered indoor scenes, providing a projective-invariant (therefore, subsuming affine invariance) approach to detect semantically meaningful mid-level features. The proposed scheme does not require learning of part models, as do existing ones to detect mid-level features. It also goes a step further compared to existing hand-crafted approaches to detecting low-level features, which only afford local affine invariance. The

method is qualitatively and quantitatively evaluated, and compared with existing representative work.

- Homogeneous texture as detected by the proposed method is shown to perform favorably in providing a crude geometric indoor layout in textured multi-planar scenes. The pros and cons are contrasted with an existing scheme that relies on computing vanishing points (ill-posed and error-prone), a simplistic Manhattan assumption, and machine learning to produce layouts.

- A pipeline is presented for indoor scene classification on the MIT Indoor67 benchmark via affine-rectified homogeneous texture detected in images. Encouraging results are obtained, which compare favorably with state-of-the-art methods, which are all learning based approaches to extracting image features. Involving deep ConvNet descriptors, the proposed approach can achieve a performance that outperforms most current state-of-the-art. The proposed approach is additionally evaluated on a set of 6200 (mostly outdoor) images, being a subset of the Places2 large scale scene dataset.

1.3 Organization of this Thesis

The remainder of this thesis is outlined below:

Chapter 2 conducts a comprehensive review of the large body of existing literature on scene recognition in general, and indoor scene recognition in particular. It also compiles the current state of the art on indoor scene recognition.

Chapter 3 presents a discourse on two possible approaches to recognizing indoor scenes, highlighting their potentials while also discussing the weaknesses and foreseeable challenges. In light of the discussion, the path adopted by this thesis is briefed.

Chapter 4 motivates the abundant presence of homogeneous texture in indoor scenes. It addresses the problem of planar affine rectification of such texture, developing a mathematical model to achieve the goal, and performing a robust estimation of instantaneous frequency and projective parameters in projected texture. The superior performance of the proposed approach over existing art in real world images marred with outliers with large spatial support, clutter and photometric severities is demonstrated via qualitative and quantitative evaluations.

Chapter 5 performs a robust detection of homogeneous texture 'in the wild', given clutter-ridden real-world indoor images. The detections are demonstrated to provide good estimates of indoor geometric layout in textured scenes, and the approach is contrasted with existing work. A quantitative evaluation of the proposed detection framework is performed, and it is seen to outperform existing representative work.

Chapter 6 presents a comprehensive set of experiments for scene classification on the benchmark MIT Indoor67 dataset, where the proposed detection framework is shown to improve performance of a number of hand-crafted as well as pre-trained deep ConvNet descriptors. Additional experiments on a subset of the Places2 large scale scene recognition dataset are also performed, further corroborating the thesis.

Chapter 7 provides a conclusion, and highlights some future avenues to exploit texture for indoor scene recognition.

Chapter 2

Indoor Scene Recognition: A Comprehensive Review

A wealth of literature exists on scene recognition, advocating novel approaches to address the problem, or tapping into various stages of the recognition pipeline to improve performance. This chapter aims to compile an in-depth survey and commentary on the literature on scene recognition in general, and indoor scene recognition in particular. Starting with a review of global and local feature based image representations in Sec. 2.1 and Sec. 2.2 respectively, a typical image classification pipeline in the context of object or scene recognition is reviewed in Sec. 2.3. Notable results from human behavioral studies found in literature are visited along the way. Sec. 2.4 surveys biologically inspired recognition, while Sec. 2.5 discusses probabilistic models. A prominent approach to indoor scene recognition — mid-level features — is reviewed in Sec. 2.6, followed by discussions on scene texture, attributes and convolutional neural networks in Sec. 2.7, 2.8 and 2.9 respectively. The chapter concludes with an overview of standard

benchmarks for scene recognition in Sec. 2.10, as well as a summary of results from recent literature on the MIT Indoor67 dataset in Sec. 2.11.

2.1 Holistic or Global Representations

A very early computational approach to fast categorization of scenes appears in [94], and attempts at quantifying certain global perceptual characteristics — naturalness, openness, roughness, expansion and ruggedness — of a given scene to obtain a so-called holistic "spatial envelope" or "gist" of the scene. These global aspects or attributes of a scene were identified as a result of trials with human participants who were asked to identify criteria they used to hierarchically divide up a set of scene images, but which should not be based on scene objects or scene semantic class. A spatial envelope property for an image is estimated by firing a corresponding pre-learned Discriminant Spectral Template on PCA bases of the DFT (quantifying the non-localized dominant structural properties, invariant of object identities and locations), or the windowed DFT of the image (characterizing localized yet holistic structural properties of the image), resulting in a low-dimensional (typically 512 features) GIST descriptor. They show that semantically similar scenes tend to exhibit similar spatial envelope properties. Furthermore, since non-localized information peforms satisfactorily (86%) for classification compared to when localized information is available (92%), they conjecture it is not necessary to first segment out regions, or identify the scene content to guide recognition of scenes. However, they only demonstrated their approach on a set of 8 outdoor scenes (albeit some being urban).

Oliva and her collaborators have since continued to argue in favour of the predictive power of global scene properties for rapid categorization, expanding upon the originally proposed handful of attributes (e.g., [44]) and sought to demonstrate that an initial scene representation need not be based on top of object recognition. However, these works have usually been limited to exploring outdoor scenes and natural landscapes, and the spatial envelope properties have been demonstrated to perform rather poorly on indoor scenes [106]. In [27], it is concluded that global scene representation such as GIST performs better in classifying the more "typical" examples of a given category. These observations suggest that global scene properties are not sufficient to quantify the "typicality" of indoor scenes. In other words, we may conclude that indoor scenes exhibit a significantly larger within-class variation as opposed to outdoor scenes.

2.2 Local Dense Features

In [31], human subjects are provided with visual stimuli involving indoor and outdoor scenes to understand various aspects of scene perception. It is revealed that at low presentation times of images (a few 10s to 100s of miliseconds), humans tend to misclassify indoor images as outdoor, but the classification is perfect at 500ms. Further, such a misclassification is not observed between natural vs. man-made outdoor images. From further experiments they conclude that this is not due to subjects possibly being able to perceive low-level sensory information or identifying objects more easily in outdoor vs. indoor images. They posit that the possible absence of perception of local cues such as edge and color due to the low presentation time might explain the bias toward labeling a stimulus as outdoor. In a

similar study [83], the authors conclude that non-localized frequency information can only help classification for a very limited number of categories, and that presence of localized primitive features such as oriented edges is necessary for recognizing most basic scene categories. They also conclude that even distinguishing between man-made and natural scene categories requires sufficient localization of primitive features. Similarly, [138] have demonstrated that densely extracted local features, such as gradient orientation histograms (i.e., SIFT [84] and HOG [19, 33]), perform better than GIST at both scene classification as well as the binary indoor-vs-outdoor classification task on their 397-category SUN database. A host of methods have been proposed that make use of densely extracted, overlapping, local features for scene classification, and are discussed in the following section.

We observe here that [95] proposed a simple 'score fusion' approach to combine SVM classifier scores for multiple features or approaches (see Sec. 2.6), and reported that local and global features when so combined yield an improved performance than either of them taken separately. Many authors have since used this fusion method to demonstrate complementarity of approaches. More principled approaches to combining GIST and local features appear in [106, 29] who propose to learn weights to fuse the two set of features.

2.3 The Classification Pipeline with Local Features

A number of steps are involved in carrying out scene classification, as depicted in Fig. 2.1. In what follows, a subsection is dedicated to the discussion of each stage in light of the research proposed in literature at that

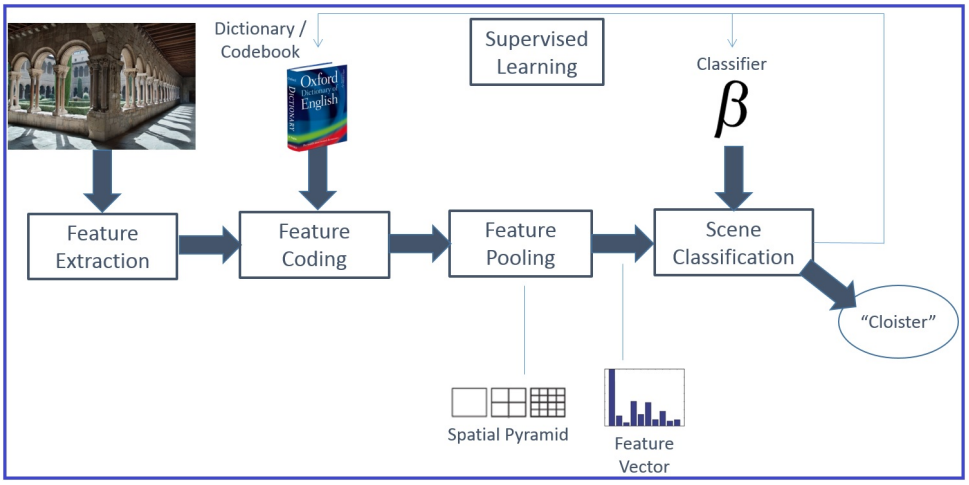


FIGURE 2.1: The various stages in a scene classification pipeline: feature extraction, dictionary learning, feature encoding and pooling, classification.

stage.

2.3.1 Feature Extraction & Description

SIFT [84] is easily the most commonly used local feature in object or scene classification approaches. Both a scale-invariant interest or key-point detector, as well as a rotation invariant descriptor were proposed in [84] to address object recognition. Key-points are obtained by searching for stable local maxima in a multi-scale difference-of-Gaussian image pyramid. The key-point is assigned an orientation(s) based on the dominant peak(s) in a histogram of weighted gradient orientations of sample points in a region around the key-point, thereby achieving invariance to rotation. The descriptor is obtained by concatenating local orientation histograms in 4x4 sub-regions from a 16x16-pixel region around the key-point. Since 8-bin histograms are employed, this yields a 128-dimensional descriptor. Local histograms provide for a local position invariance in the descriptor. Further,

suitable normalization of the descriptor is performed to achieve invariance to affine illumination changes, and to reduce effect of non-linear illumination changes. Note the SIFT detector is not affine invariant, though it has been shown to be resilient to affine distortions or 3d viewpoint changes [84].

A prominent affine invariant interest point detector in literature is the MSER [88], proposed in the context of wide baseline stereo correspondence, but which has also been successfully employed for image retrieval [120]. It detects blob-like regions of high contrast w.r.t their surrounding. Another affine-invariant detector to note is the scale-saliency detector of [64], which extracts blob-like regions that are salient in the sense that they exhibit unpredictability in their local attributes and over spatial scale. A notable property of this detector is its intra-class invariance which led to its wide use in object recognition [35, 30, 26]. Repeatability under intra-class variation is also a highly desirable property for scene recognition, as corresponding regions or parts in similar scenes often possess large amount of intra-class variation. For example, two dining rooms can and do contain chairs of different shape and color. The early probabilistic scene model of [32] demonstrated a slightly improved classification performance by the scale-saliency detector over the SIFT detector. However, it also demonstrated that dense sampling of local SIFT descriptors provides a substantial improvement over local sparse interest-point based description of a scene, and this is corroborated by the contemporary work of [6]. Interestingly, where local sparse features perform very well on scene retrieval [120] (which is the problem of retrieving all scene images from a database the *same* as the query, but possibly varying in photometric or geometric properties), they perform very poorly on scene recognition (which may be regarded as the problem of establishing correspondence between two images depicting a *semantically similar*, but not necessarily the *same* scene). Consequently, sparse feature

description is altogether non-existent in scene recognition. More recent works of [139, 138] seeking to compare various description approaches have also demonstrated the low performance of sparse SIFT, and it is commonly understood that a sparse image description is less discriminative compared to a dense description.

Another popular local image descriptor is the HOG. Originally proposed by [19] for pedestrian detection, it has since become the standard descriptor for generic object detection [33] due to the reason that it provides remarkable detection performance with only a linear SVM classifier. The image at a given scale is divided into 8x8-pixel non-overlapping cells. Each cell yields a 9-bin gradient orientation histogram aggregated over the cell region, which is contrast-normalized 4-fold by the gradient energy in the four blocks covering that cell (blocks being overlapping 2x2-cell regions). Hence, each cell yields a 36-dimensional feature vector. Vectors from spatially neighboring HOG cells may be concatenated to describe a larger object at given image scale. Analysis for pedestrian images in [19] reveals that a linear SVM detector learned over HOG features is able to cue on discriminative gradients while rejecting gradients that exhibit high intra-class variation. The authors in [33], based on empirical analysis, concluded that the top 11 eigen vectors of HOG not only capture all the information but also lie in a linear subspace defined by 13 sparse vectors, each 36-dimensional. This analysis led them to propose a 31-dimensional variant of HOG which preserves performance of the original version. Recently, authors have also employed concatenated vecotrs from 2x2 HOG cells for scene image representation [139, 138, 55], demonstrating a moderate improvement compared to SIFT. Local dense HOG has also been used for object recognition [133].

Since descriptors such as SIFT and HOG are not inherently scale-invariant [48], spatially overlapping image patches on a regular grid are extracted

at multiple scales before dictionary learning and feature encoding for a more scale-invariant overall description of the image. At the same time, local extraction of features provides spatial invariance and robustness to occlusions compared to a global descriptor such as GIST [94] (see Sec. 2.1).

Finally, for the sake of completeness, we note that comprehensive surveys and quantitative evaluations of local affine invariant detectors and descriptors may be found in [90] and [89], respectively, and would be of interest to readers working on problems of object recognition or scene retrieval. The brief review performed in this section, however, was geared more toward scene recognition, and based on the more recent literature on the problem.

2.3.2 Feature Encoding

Encoding is the process of representing local image features in terms of a dictionary of codewords, textons or atoms. The earliest encoding scheme is probably the bag of words (BOW). BOW has its origins in text document retrieval, and was introduced into computer vision by the pioneering work of [120] for scene retrieval. The simplest procedure involves clustering descriptors extracted from a training set — either sparsely or densely — via K-means into a dictionary or codebook of representative codewords. This is called *dictionary learning*. Now features from any given image are vector-quantized to one of the dictionary atoms, and a histogram of dictionary atoms so obtained is called a bag of visual words, or simply a bag of words representation of the image.

Note that any spatial ordering or local co-occurence relationship between features in an image is lost in this approach. A seminal work attempting to preserve some degree of spatial ordering into the bag of words scheme is that

of [70], which put forth the now widely popular scheme of spatial pyramid matching (SPM). The scheme borrows the idea from Pyramid Match Kernel (PMK) [43] which was proposed for feature space. In SPM, the same approach is applied to the 2D image space, while performing traditional clustering and vector quantization (as in BOW) in the feature space. The process involves partitioning the image into increasingly finer sub-regions, and obtaining a separate histogram (BOW) for each region. The number of regions depends on the number of pyramid levels. At the lowest level, only one regon exists (the entire image). At the second level, the image is divided into 2x2 sub-regions. At the third level, the image is divided into 4x4 sub-regions, and so on. For a 3-level pyramid, therefore, we obtain 21 cells. For a dictionary size of, say, 200, a concatenation of all region-specific histograms yields a 4200-dimensional image representation. Classification is performed via SVM employing a histogram intersection kernel (HIK). Conceptually simple and computationally efficient, [70] exceeded state-of-the-art performance on the object recognition dataset Caltech101 [30], extended the prevalent 13-class scene dataset [94, 32] to 15 classes, and defined the state-of-the-art on this testbed. Indeed, evidence from human behavioral studies suggests that both local, region-based as well as global, configural information is required for more effective classification [130]. The SPM approach was also extended to 3D in [45] for categorizing video scenes. [96] allowed image spatial sub-regions to be reconfigurable and take on any of a set of region models, thereby generalizing the SPM framework which works with fixed region models. The recent work in [140] trains a model to predict planes and their 3D orientations in single image indoor scenes, and uses these orientations to define pooling regions for features. Combined with SPM, the work achieves a state of the art performance on the MIT Indoor67.

The BOW approach assigns an image feature to a single dictionary codeword. This is known as hard quantization in literature. Soft quantization generalizes it by allowing multiple codewords to linearly combine in order to reconstruct the image feature and minimize reconstruction error [103]. One such approach is sparse representation (SR), which, previously having found application in face recognition, image restoration and motion segmentation, was introduced to object and scene classification by [141]. Here sparsity is enforced on the coefficients of the linear combination, essentially allowing only a very small subset of the dictionary atoms to reproduce a given feature. It was also demonstrated by [141] that a linear SVM is sufficient to classify sparse coded images compared to kernel SPM. However, a substantially large (overcomplete bases) codebook size (1024) is needed compared to the 200 by kernel SPM to preserve performance.

In [39], the authors point out that due to the overcomplete nature of the codebook, and the independent encoding process of each feature, features that are similar end up being represented as widely varying sparse codes. They propose Laplacian sparse coding which adds another term to the sparse coding objective to force similar features to possess similar sparse codes. The approach is shown to substantially outperform both SPM and SR. A related work is [133], though it deviates from sparse coding. They linearly encode each feature in terms of its K-nearest neighbours in feature space in the dictionary. Named Locality constrained Linear Coding (LLC), the process in essence performs feature selection by selecting local bases for each descriptor to form a local coordinate system. It is pointed out that locality is more essential than sparsity, as locality necessarily leads to sparsity but not vice versa. No experiments are reported for scene, however. The approach in [144] starts at the raw pixel level, rather than employing SIFT-described patches, and uses a 2-layer hierarchical scheme

for sparse encoding raw pixel patches. The idea is that since a data-adapted sparsifying dictionary is already learned in the process, one might as well employ raw pixel patches. The approach — though not applied to scene — outperformed prevalent object recognition methods on Caltech101. [146] is another representative sparse coding approach demonstrating high performance on the 15-category scene dataset. They observe that since a max-pooling stage follows sparse coding, allowing the sparse coefficients to take on negative values is detrimental. They force sparse codes to be zero or positive, and also employ low-rank decomposition of resulting image representation to reject non-representative scene features as sparse noise.

Another soft encoding method is the Gaussian Mixture Model (GMM) [26, 100] (learned via Expectation Maximization) which models both the deviation of patches from cluster means as well as covariance. Other popular methods are the Fisher encoding [101] which captures first and second order differences between the image descriptor and the centers of a GMM, and Super vector encoding [11] where only first order differences are computed, besides considering the cluster mass, and normalizing each cluster by the square root of the posterior probability rather than the prior (as is the case in GMM). Comprehensive surveys and guidelines on best practices for various feature encoding methods for classification appear in [11, 54]. Empirical analysis by [11] on the object recognition benchmarks Caltech101 and PASCAL VOC 2007 reveals Fisher encoding with Hellinger kernel (as well as Super vector encoding) to perform better than other encoding schemes such as hard / soft (e.g., LLC) quantization, even when these approaches employ non-linear kernels such as the Chi-squared. Hence, encoding higher-order differences between the descriptor and the codewords seems to compensate for information otherwise lost due to quantization. The superiority of Fisher encoding over other schemes holds for indoor scene recognition as

well, as corroborated by [63].

2.3.3 Dictionary Learning — Unsupervised

K-means is used to generate a codebook for hard quantization (BOW), while GMM for Fisher encoding. However, the sparse coding approach requires a computationally expensive process of learning the *sparsifying* codebook, one that best approximates each training sample under certain sparsity constraints. This is a non-convex problem, hence iterative approaches are employed. A pioneering approach is the K-SVD [1], a generalization of the K-means algorithm. The dictionary and sparse coefficients are updated iteratively. What makes the approach different and faster from others is that when updating a dictionary atom, its corresponding coefficients in the sparse representations of all data vectors are updated as well. In this sense, it is a more direct generalization of K-means, as each dictionary column is updated separately (via SVD) as done in K-means.

Another dictionary learning algorithm is that of [75], which is also employed by [141] as it is considerably more time-efficient than previous approaches. Fixing the sparse coefficients of the training samples, they propose to solve the problem of optimizing the objective over the dictionary bases via the Lagrange dual formulation, and show this requires significantly fewer optimization variables. To optimize over the sparse coefficients, a 'feature sign search' algorithm is used, wherein signs of the coefficients are guessed rendering the quadratic programming as unconstrained and efficiently solvable. The algorithm is demonstrated to also replicate certain phenomena observed in neuroscience i.e., end-stopping and surround suppression, previously unexplained by linear models. This is because sparse coding is a non-linear process where bases compete to best represent the image and

maximize the sparseness, hence it can effectively model inhibition between bases (neurons). Another prominent work on codebook learning is [145], which attempts to learn a block-sparsifying dictionary whenever a block-sparse structure exists in the data under consideration. No prior knowledge on the subspace membership of signals is required; the underlying block structure is automatically recovered.

A notable work, particularly relevant to scene recognition, is [134], which employs sparse representation of covariance matrices, achieving good performance on scene recognition. A probabilistic generative model is used to jointly learn — in a maximum likelihood (ML) setting — a dictionary to linearly (no sparsity) encode patches, as well as a dictionary of positive definite covariance patterns to sparsely encode regions (consisting of a number of patches). Given the dictionaries, inference for the representation of patches and regions is performed in a MAP framework. The generative model is approximated via a coordinate-wise convex optimization scheme. The motivation is based on the observation from a work in computational neuroscience [65] that a given scene region exhibits a characteristic pattern of covariance among the features encoding individual patches in the region. Hence, regions can be encoded via their region covariance, and [134] proposes to infer sparse representations of region covariances in terms of a ML dictionary of covariance patterns. One notes that region covariance has also been independently proposed in computer vision as a feature descriptor for image regions for detection and classification [127]. A few strengths of the covariance SR approach may be identified. Considering that the feature learning framework only starts with vectorized raw pixel values in 5x5-pixel patches, a rather robust and discriminative image representation is learned. It is shown that only linear kernel is required to achieve good performance. This is in agreement with previous SR work

[141] and strengthens the general observation that sparse representations are linearly more discriminative. By contrast, non-SR works require non-linear kernels, such as HIK [70] or Gaussian [139]. At the same time, a few weaknesses of the covariance SR approach should be noted. Firstly, this approach requires a dictionary 4 times as large (4096 atoms, 16x16x4096/2 = 524,288 values to be estimated) compared to the one employed by [141] (1024 atoms, 1024x128 = 131,072 values to be estimated) to achieve a comparable performance. Since the number of regions that need to be sparse coded is much lower than the number of patches to be encoded in [141], it suggests that perhaps covariance matrices are not conducive to sparse representation. No experiments are reported in [134] that consider the effects of varying dictionary size. Furthermore, since dictionaries learned on one dataset (Scene15) are able to generalize well to another, very different dataset (Indoor67), it may be argued that learning a dictionary may not be relevant and that it would be better to use pre-defined non-learned bases.

2.3.4 Dictionary Learning — Supervised

Supervised sparsifying dictionary learning has been studied in the context of applications such as face recognition, handwritten digit recognition and texture classification [136, 86, 85, 143]. In face recognition, the dictionary merely consists of the training face examples. An incoming test face image tends to have non-zero coefficients only for the dictionary atoms corresponding to its class when sparse coded [136]. This is because aligned face images are known to roughly reside in a low dimensional subspace. For applications such as texture or digit recognition, one naive approach is to train a separate dictionary for each class [86, 85]. At test time, the dictionary that minimizes the reconstruction error for a given patch defines its class. In [85], the authors point out that this naive approach is

essentially reconstruction based. They propose to also employ the residual errors of a patch given each of the class-specific dictionaries to model the fact that a class-specific dictionary should be good at reconstructing that class but bad at reconstructing other classes, thereby introducing a class-discriminative constraint. The work of [7] also attempts to add discrimination criteria to the basic reconstructive technique. Termed Fisher Discrimination Dictionary Learning (FDDL), the approach adds the Fisher discrimination criterion (similar to Linear Discriminant Analysis) into the dictionary learning formulation. Specifically, since the final classification is based on the sparse codes of the patches, they propose to minimize the within-class scatter of the sparse codes, and maximize the between-class scatter.

One notes that these dictionary learning schemes are defined for applications which classify individual patches (say, 32x32 pixels) depicting faces, texture or handwritten digits. Adapting dictionaries to training samples of this kind of data, and enforcing discrimination criteria on the sparse codes makes sense for these applications. In generic object or scene recognition, however, one deals with a lot of densely sampled, overlapping patches from the image (say, 700 – 900 patches at 16x16 pixels from a typical 480x640 image). A global vector representation of the object or scene image is obtained after encoding these patches (say, via SR), and then pooling the codes over spatial bins. Hence, classification is based not on the patch representation, but on the final global image representation. Therefore, these dictionary learning schemes, which work at the patch level, cannot be expected to perform well for object or scene classification. A work [110] questioning the relevance of sparse representations for generic image classification concluded that sparsity is not necessarily required for classification, but might be important when learning the filters (bases). These results seem to be in

line with our observations above.

At least three works may be identified that have addressed this gap, and proposed solutions [7, 79, 142]. While differing slightly in their mathematical details (the loss function employed, the method of differentiation employed, etc), all these schemes unify the process of dictionary learning and learning the classifier by jointly minimizing the classifier loss function over these parameters (i.e., the dictionary and the classifier weights), thus attempting to learn a more conducive dictionary that lends more class-discriminative global image representations. From these works, however, mixed findings for scene and object recognition can be observed, possibly due to the fact that the overall problem formulation is non-convex and the solution susceptible to initialization.

2.3.5 Feature Pooling

Unlike hard quantization (BOW), soft encoding does not directly result in a single image descriptor, and feature pooling must be performed to obtain an overall image representation. This is the processing of combining the responses to a basis atom for all the patches in a given image region via a sum, average, max or some other function independent of the spatial order of the contributing bases. In this way, pooling attempts to achieve some degree of local invariance over position (and scale, if provision is made, as in the biologically inspired hierarchical HMAX models [116, 91]). Some theoretical and empirical analysis of feature pooling appears in [7]. One observation in [7] is that max pooling substantially improves linear classification performance irrespective of the coding module. Furthermore, it

is revealed that the worst-performing coding scheme (hard vector quantization) paired with max pooling outperforms sparse coding paired with average pooling. A more detailed theoretical treatment is given in [7].

One approach to improve recognition performance via the pooling stage is that of [60]. The key idea is to learn optimal image regions over which pooling is performed. This is in contrast to the traditional SPM framework [70, 141], where a set of pre-defined 'receptive fields' (regular grids at multiple pyramid levels) are employed. The problem of learning these optimal receptive fields for pooling is posed as one of performing feature selection on a high-dimensional vector, which results from pooling over each codebook atom over each of an over-complete set of receptive fields. In this manner, the scheme 'selects' the most relevant combinations of codebook features and over-complete receptive fields. Object recognition performance is comparable to the state-of-the-art, however.

Another scheme aiming to improve recognition performance by tapping at the pooling module is [34]. Targeting single-object image classification (Caltech101), a pooling operator - different from average or max - is formalized i.e., a weighted Lp-norm. It enforces two constraints. Firstly, the between-class variance of the k-th pooled feature (k-th visual word) is maximized, while the within-class variance is minimized. Secondly, smoothness constraint is enforced on the weights, called geometric coefficients, which encode the contribution of the m-th image location for the specific visual word. For a given feature k, the geometric coefficients for adjacent spatial locations are constrained to be similar. In single-object images, this constraint leads to having the coefficients for the feature on the object to have similar values, while the coefficients not on the object to have lower values. The approach defined the state-of-the-art on Caltech101, considerably outperforming all prevalent approaches. Improvement is also demonstrated on

the Scene15 dataset.

The study in [38] showed that image saliency may be used to define two disjoint and equal pooling regions in a scene — a salient region and a non-salient region — that are not spatially biased (unlike SPM [70]). While good performance is reported on Indoor67, a direct comparison with SPM with the same experimental setup is not provided. [9] proposed to learn weights in the image grid for nearest neighbor distances based on the spatial layouts of visual words in training images. Improvement was demonstrated for Indoor67 over the baseline nearest neighbour method.

2.3.6 Classification

All best-performing and state of the art scene classification methods employ the Support Vector Machine (SVM) classifier. Since SVM is a two-class classifier, one of two approaches is used for multi-class classification. The more popular approach called one-versus-all, or one-versus-rest, trains N SVMs, one for each class, treating examples from all other classes as negatives. Another approach, called one-versus-one, rarely seen, learns $N(N-1)/2$ pairwise SVMs, and chooses the class for a test example which is selected by the most classifiers. Depending on the local descriptor or encoding scheme used, linear [141, 133] or non-linear kernels [70, 11, 138, 137] are used. Probabilistic models [96, 32] employ Bayes classification. The non-parametric KNN classifier — more popular in texture classification — is rarely seen [6, 9], because, as revealed in [5], they lose their capability when descriptor quantization and image-image distance metrics, common in object or scene classification, are used. Experimental analysis in [139] corroborates the low performance of nearest neighbor based scene classification as compared to SVM.

2.4 Biologically Inspired Recognition

A line of work on visual recognition somewhat different from the pipeline reviewed above (but having many similarities) is based on the HMAX model [109, 115]. This is a hierarchical model, wherein multiple layers are employed with the aim to mimic the processes and invariance properties of the simple and complex cells in the primary visual cortex (also called the striate cortex or V1) in primates. A representative work [116] adopts the model for large scale real-world object recognition, comparing favorably with contemporary indigenous computer vision systems on testbeds such as Caltech101. Essentially, feature computation is carried out over a number of layers, starting with image responses to oriented Gabor filters over a multi-scale pyramid. Subsequent layers achieve local invariance via max pooling, further filtering via prototype features (which may be likened to a dictionary of features), and another global pooling stage for each prototype. Classification is performed via SVM.

The work [91] further proposed some biologically inspired improvements over the base model. Notably the employment of sparse prototype feature vectors, mimicking the cortical phenomenon of lateral inhibition and the limited receptive fields (pooling regions) of neurons in the higher visual areas V4 and IT. Lastly a feature selection method is also used in conjunction with SVM for classification. The proposed modifications considerably improve classification performance over the base model for Caltech101, albeit still outperformed by the purely computationally approaches. [58] perform empirical investigation in further detail for learning dictionaries based on the models in [116, 91], while [62] learn an overcomplete dictionary with non-negative sparse coding (see [146]) of features for the HMAX model, demonstrating an improvement over [141] for scene classification.

The model of [65] aims to learn statistical distributions (region covariance) that characterize local image regions and identify them from individual image patches. This allows the neural code to represent more abstract aspects of the image and remain invariant across fixations within local regions. Model parameters are learned by maximizing the likelihood of the train images under the model, and the response of model neurons to a given patch is obtained as the most probable neural representation by maximizing the posterior probability. The region covariance is a function of the neural activity. The proposed model is shown to exhibit cortical neural properties such as phase invariance, orientation tuning and complex suppressive effects, and inspired the work in [134].

2.5 Probabilistic Models

The semi-supervised (since only per-image class labels are available, theme labels in an image are not given) probabilistic framework of [32] introduces latent variables to learn a distribution of intermediate abstract scene properties called themes, which they liken to textural properties. However, a poor performance on indoor as opposed to outdoor scenes is demonstrated. A similar but unsupervised approach, appears in [6], where a generative model — probabilistic Latent Semantic Analysis — is adopted to automatically discover 'topics' in images (which may be objects or scene regions), obtain a distribution of the discovered topics for a given image, and use this description with non-parametric nearest neighbour classification. [77] is a hierarchical generative model that jointly recognizes and segments scene object components as well as classifies the overall scene.

An interesting generative approach appears in [96], which describes a scene region by the region model that maximizes the posterior probability of

such an assignment (MAP). The model parameters are learned via Maximum Likelihood Estimation (MLE) using Expectation Maximization (EM). A discriminative counterpart of the approach is also presented which is learned using a latent (since assignment of region models to scene regions is not known in training images) structured SVM. They show that initializing the discriminative model with parameters obtained by EM improve performance of the discriminative version of their framework. A performance on indoor scene recognition is reported that is comparable to the then-state-of-the-art.

2.6 Regions-of-Interest, Parts or Mid-Level Features

In the seminal work of [106], the authors demonstrated that contemporary global [94] or local dense features [32, 70] popular in scene recognition at the time did not perform as well on the indoor subset of categories in the 15-category dataset as on the outdoor subset. The underlying reason is the inherent presence of much larger intra-class variability in indoor scenes as opposed to outdoor scenes. (One notes the later work of [139], having conducted a performance comparison of local dense feature descriptors on an even larger indoor+outdoor dataset, also report higher outdoor classification performance as opposed to for indoor, while performance for urban scenes comes in third). In order to investigate and address the problem further, [106] collected a new large-scale dataset consisting of 67 indoor scene categories. They assume a set of 'prototype' unlabeled but human-annotated or automatically segmented images is given, where segments are *regions of interest* (ROIs) that depict objects or semantically meaningful regions in a scene. They then learn per-class parameters to minimize the

FIGURE 2.2: Prototypes for two Indoor67 categories (`church_inside` and `inside_bus`), sorted by their weights. First 7 columns correspond to highest ranked, while last 2 columns the least ranked prototypes for shown category. Thickness of ROI's bounding box is proportional to its weight. Adopted from [106].

distance between the prototype ROIs, and segments in the train images of the class, thereby also learning a prototype's weight for a given class. Intuitively, the model aims to determine what ROIs can typically occur in a given scene category. Fig. 2.2 shows prototypes and ROIs for two selected categories from Indoor67: `church_inside`, the best performing category with 63.2% via their model, and `inside_bus` at 39.1%.

The model outperforms GIST (21%) on this dataset, but the accuracy is still very low (26.5%). The reason may likely be attributed to their use of bag-of-words representation for image ROIs and segments based on sparse (and not dense) SIFT. Nevertheless the work of [106] drew considerable attention from the computer vision community toward ROI based indoor scene recognition, and ensuing approaches employing object detection-style HOG features over a grid of neighbouring cells demonstrated significant performance gains. Specifically, [78] aimed to leverage the availability of a number of annotated object datasets, such as LabelMe and ImageNet [21], in order to train 200 full-blown object detectors. Detectors for structured

objects such as tables are trained via the *deformable part model* (DPM) of [33], while an existing texture classifier is employed to detect image regions with textural and material properties. Multi-scale object detection is run on an image to obtain detector responses, and pooling is performed over all image scales within a spatial pyramid bin for a given detector, to obtain an overall image descriptor for classification. An impressive performance is reported (37.6%), though one should note that pre-trained object and texture detectors have been employed.

Another work proposing to use *part modeling* for indoor scene recognition is [95]. It essentially approaches the problem as that of scene detection via a DPM [33] learned for each class. Specifically, a large, coarse-scaled HOG 'root' detector fires on the the entire scene. Eight smaller, fine-scaled part detectors, that are deformable with respect to the root, fire on characteristic regions or objects in the scene. In practice, a 2-component mixture model per scene category is trained to cater to images having different viewpoints. Learning is performed using the latent SVM formulation of [33]. Fig. 2.3 illustrates two part models and sample detections.

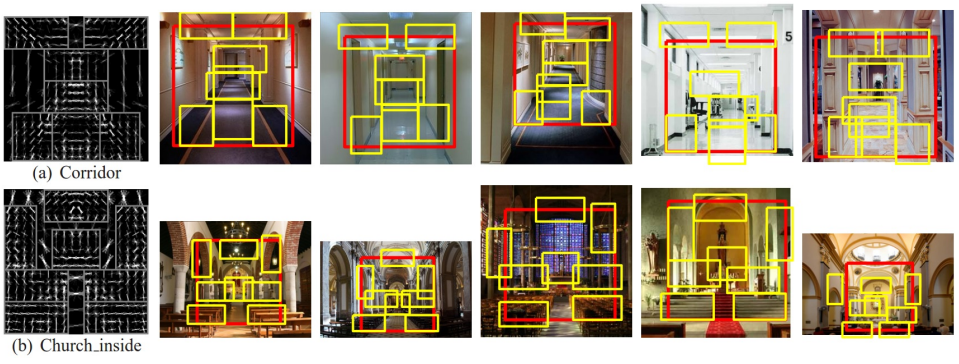


FIGURE 2.3: Scene DPMs and sample detections for two Indoor67 categories (`corridor` and `church_inside`). Adopted from [95].

The approach achieves an average accuracy of 30.4% on Indoor67. To leverage the complementarity with DPM of other feature representations,

such as GIST and spatial pyramid (SP), the authors proposed the following 'score fusion' scheme. Each representation (GIST/SP/DPM) yields n one-vs-all classifier scores, one for each of the n scene classes. Let a_i be the score on a test image for the i^{th} classifier from one of the representations. The softmax transformed score is therefore $\frac{e^{a_i}}{\sum_{k=1}^{n} e^{a_k}}$. The confidence for class i is then given by a multiplication of the softmax scores of all representations. By score fusion of GIST, SP and DPM, [95] achieved an average classification performance of 43.08% on Indoor67, the state-of-the-art at the time.

A similar, more recent work is [81]. Instead of penalizing part scores based on their deviation w.r.t the root, a different approach to model part locations is taken. Part scores are modulated via Gaussians modeling clusters of part locations (named 'spatial pooling region') in normalized image space. An Indoor67 performance of 50.1% is reported, and combined with Fisher encoded dense SIFT, the approach achieves 68.5%.

A very interesting work is presented in [119, 24], where the goal is to automatically mine representative (frequently occurring), and discriminative patches for a given scene class (or city in [24]) in an unsupervised manner. Since there is no supervision, the patches can correspond to objects, parts of objects, or larger representative image regions, but are not constrained to be any one of them. They term such features as *mid-level* visual features or primitives, and conjecture that they are better at generalizing to similar instances exhibiting large intra-class variations than do low-level features. The mining problem is posed as that of discriminative clustering — an iterative approach that alternates between clustering (essentially, running SVM detectors over a multi-scale image HOG pyramid), and training a discriminative classifier (SVM) for each cluster. The key novelty of [119] lies in employing careful cross-validation between iterations. Specifically, one

training subset is used to train detectors that are fired on another subset to obtain positives for refining or retraining the clusters. The idea is to avoid SVM overfitting. Iterative hard negative mining is employed to handle the large number of negative images, as in [33]. Fig. 2.4 shows representative clusters for some Indoor67 categories obtained by applying this approach in the course of experimental work for this thesis. The method can be observed to produce clusters with surprisingly good visual consistency for a fully unsupervised approach, and detectors appear to exhibit remarkable intra-class invariance.

FIGURE 2.4: Representative mid-level feature clusters obtained for sample Indoor67 categories by applying the method of [119] (clockwise): `church_inside`, `cloister`, `corridor` and `inside_bus`. Three clusters are depicted for each scene category.

[119] applied the method to Indoor67, mining 210 clusters per category. When training the scene classifier for a given category, the image descriptor is a 1050 dimensional vector obtained by max pooling over the response

map on a 2-level spatial pyramid using the detectors for that category. Final classification is done by using softmax transformed scores (since the feature set for each classifier is different) of the 67 one-vs-all classifiers. The approach scores a 38.1% on Indoor67. Using the score fusion technique of [95], a classification accuracy of 49.4% was reported, the then-state-of-the-art. [24] employed the approach to discover visual elements that are geographically informative for a given city locale.

The experimental work of [63] demonstrated a significant improvement over [119] in both classification performance and speed. They employ a super-pixel based part seeding procedure, incrementally evolve a part detector starting with one positive training example to train an exemplar SVM, and subsequently use the LDA classifier of [46] instead of SVM detectors. Finally, 50 informative parts per scene category are short-listed via an entropy based ranking method, that are distinctive for the given category but may also manifest in a few of the other categories. The LDA approach essentially computes a detector as the difference between the average positive and negative features in a 'whitened' HOG space. The whitening transform and negative mean may be computed once for the entire dataset, foregoing the need to perform a computationally expensive hard negative mining [119, 33] process every iteration, thereby accelerating the process multi-fold. [63] reported a 46.10% accuracy on Indoor67, and, combined with Fisher encoded dense SIFT, achieved 63.10%. Another, principled, approach to part mining is [23], who propose a discriminative variant of the mean-shift algorithm, maximizing the density ratio, to obtain representative and discriminative scene parts. Using 200 parts per category, an Indoor67 classification performance of 64.03% is reported, and, combined with Fisher encoded dense SIFT, 66.87% is achieved.

2.7 Texture

A local texture descriptor proposed specifically for scene recognition appears in [137]. Named CENTRIST (CENsus TRansform hISTogram), the method captures occurrence histograms of local structure in images. A Census-transformed image is first constructed, wherein a 3x3 pixel neighbourhood is thresholded by the value of the center pixel to obtain a binary code. The approach is similar to the LBP (see Sec. 6.2.2, [93]), except instead of weighting and summing up the resulting bits over the neighbourhood, CENTRIST assigns the entire 8-bit binary code to the center pixel. A 256-dimensional histogram of such 8-bit codes may be extracted for a given image patch, serving as the descriptor. Like LBP in its basic form, CENTRIST is invariant to monotonic photometric changes, but is sensitive to rotation. It is demonstrated in [137] that while SIFT can assign image patches depicting similar visual structure to different codewords, CENTRIST tends to assign them to a common codeword, indicating that CENTRIST can better generalize to similar instances. Reducing CENTRIST to 40 dimensions via PCA, and employing a spatial pyramid representation yields a so-called sPACT (spatial PCA of CENTRIST) representation. The then-state-of-the-art performance of 36.88% was reported on the MIT Indoor67.

Another approach to texture-based scene representation is that of [87]. Motivated by the fact that different features tend to exhibit different dominant orientations, and that descriptors in scene recognition should *not* be rotationally invariant, they propose to extract information from a given patch at multiple orientations. A given $N \times N$ patch is divided into N strips oriented in a given direction. Each point on the corresponding 'oriented texture

curve' is essentially the mean of pixel values along a given strip. A discriminative, illumination and geometric invariant curve descriptor is proposed by making use of curve gradients and curvatures, followed by a normalization step to overcome local contrast changes and suppress texture-less patches. Descriptors sized 185 features are extracted from dense patches sized 13x13 pixels. Considering only a simple bag-of-words encoding is used on a 3-level spatial pyramid, a good performance of 47.33% is reported on the MIT Indoor67.

2.8 Attributes

[98] presented a large scale scene attributes database built on top of the SUN dataset [139, 138]. 14,000 images form 700 categories were annotated with 102 attributes by crowd-sourcing on Amazon Mechanical Turk. The attributes contain functional/affordance based (e.g., camping, sailing), material (e.g., vegetaion, glass), surface (e.g., moist, rusty), and spatial envelope ([94], Sec. 2.1) properties. However, the analysis presented indicates that although recognizing attributes from a feature-rich representation (GIST, HOG2x2 and other features) is quite feasible, scene classification even from human-annotated attributes has a very low performance. This suggests that an attribute set consisting of 102 properties may not afford sufficient discriminative power for classification.

2.9 Deep Convolutional Neural Networks

Introduction. Convolutional neural networks (CNNs or ConvNets) [71, 72] are artificial neural networks (ANNs), that have successfully been used

in various machine learning applications including, but not limited to, computer vision [74, 107, 15, 57]. As in conventional neural nets (multi-layer perceptrons, MLPs) [92], CNNs also contain fully connected neuron layers, but these appear at the end of the network. Preceding these fully connected layers, CNNs additionally feature layers where the neurons (nodes) are essentially convolution operations on overlapping, tiled regions in the input, naturally lending themselves to processing images. Depending on the particular network architecture, max-pooling layers appear after some convolutional layers, serving to downsample the spatial resolution while providing local positional invariance. All layers may typically be followed by ones applying an element-wise non-linear activation function. In classification settings, an N-way soft-max layer serves as the output layer. Modern CNNs are "deep", consisting of tens of hidden layers (i.e., non-input or non-output layers), featuring hundreds of thousands of neurons with tens or hundreds of millions of parameters so as to achieve a large-scale learning capacity [68, 118, 125, 50].

Like the HMAX model (Sec. 2.4), CNNs are biologically inspired by the visual cortex; the early layers attempt to mimic simple cells, which have a limited receptive field, responding to local, edge-like patterns. Subsequent layers model the behaviour of complex cells, capturing higher-level visual structure and patterns by examining information over larger regions in the input space, and progressively evolving a more and more abstract and invariant representation. Unlike the HMAX model, however, CNNs, being a variant of ANNs (also inspired by the biological neural network), automatically learn the network parameters (the neuron weights, i.e., filter banks and those in the fully connected layers) from training images via stochastic gradient descent. In doing so, an algorithm called backpropagation is used for the fast computation of the cost function's gradient, and requires the

activation function be differentiable [72, 92].

Deep Learning. Deep learning [42], i.e., the training process of deep (multi-layered) ConvNets requires massive amounts of labeled data, and is understandably computationally very demanding. In recent times, however, the availability of huge repositories with millions of hand-labeled images [21], as well as that of massively parallel computing power such as distributed clusters or general-purpose GPUs, have contributed toward the practical realization of high-performance deep CNN systems. Consequently, deep ConvNets have taken the computer vision community by storm, especially proving to be highly successful in solving, among other tasks, large scale recognition problems where conventional approaches have struggled [113]. The phenomenal success can be attributed to the highly discriminative and relevant hierarchical features a deep network can automatically discover due to the layered structure of the model.

Beginnings. ConvNets were first introduced in [71, 72] in order to exploit the unified feature extraction and classification learning paradigm of conventional multilayer neural nets, while additionally incorporating domain-specific priors for the task of image classification. Specifically, the fully connected layers are relegated to the final stages of the architecture, and convolutional layers are introduced to ensure local receptive fields, shared weights and sub-sampling. This not only drastically reduces the number of connections (and, hence, parameters to learn) compared to a fully connected architecture of the same size, but also provides shift or translational invariance. Known as **LeNet-5**, this early model featured 3 convolutional layers, 2 sub-sampling layers, a fully connected layer and an output RBF layer, making for a 7-layered architecture with 60,000 free parameters to learn (albeit having 340,908 connections) [72]. It was demonstrated to outperform methods using hand-crafted feature extraction in the context of

handwritten characters and bank check recognitions. Studies have followed that show the superior invariance provided by CNNs for generic object recognition [73], and that empirically investigate and compare various configurations of the CNN architecture [59].

Contemporary Architectures. More recently, a high-performance convolutional network was trained by [68] on a subset of the ImageNet dataset containing 1.2 million hand-labeled images of 1000 object categories, as defined by the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [113]. Now widely known as **AlexNet**, this CNN architecture features 5 convolutional and 3 fully-connected layers, followed by a 1000-way softmax layer for classification. Novel additions to the architecture include non-saturating Rectified Liner Units (ReLUs) as the non-linear neuron activation function, which is shown to significantly speed up the training, Local Response Normalization (LRN) to improve generalization, and Overlapping Pooling to reduce overfitting. A pair of high-performance GPUs were leveraged, allowing to train a larger network. Overall, the architecture contains 60 million trainable parameters. The model is shown to provide record breaking performance improvement over the best results on the challenging ILSVRC-2010 and ILSVRC-2012 tasks.

[118] perform an empirical study wherein the depth of the network is increased from 11 through to 19 layers. Known in the community as **VGG-VD**, they demonstrate that such "very deep" architecture configurations are possible to train since they employ the smallest possible filter kernels (i.e, 3x3). Moreover, max-pooling is performed only after every 2 or 3 stacks of convolutional layers, resulting in effective receptive field sizes of 5x5 or 7x7, respectively. As opposed to other models, e.g., AlexNet, the convolution stride is also reduced to 1 pixel, which is computationally feasible owing to the small kernel size. The number of free parameters is 138

and 144 million, respectively, for their 16 and 19-layered networks. The architecture won the 1st and 2nd places in the localization and classification tracks at ILSVRC-2014.

Other notable CNN architectures include **Caffe** [61], the 22-layered **GoogleNet** [125] (which achieved the 1st place in ILSVRC-2014 classification task) and **OverFeat** [114] (participant in the ILSVRC-2013 challenge). Most recently, **ResNet** [50] attacked the degradataion problem in very deep networks, wherein the training accuracy saturates and then falls drastically, by learning residual functions with respect to layer inputs. Their ultra-deep, 152-layered architecture won the ILSVRC-2015 classification, detection, as well as Microsoft COCO-2015 detection and segmentation challenges.

CNN Features as Off-the-Shelf Descriptors for Scene Recognition. DeCAF [25] demonstrated that features learned on the large and diverse ImageNet subset by [68] can successfully generalize as off-the-shelf features to a number of tasks such as scene and fine-grained object recognition. Such domains typically feature limited train data, on which huge architectures such as that of [68] are likely to overfit. The empirical study of [25], however, has demonstrated that generic deep features trained on a fixed but huge dataset can not only generalize to other domains, but also significantly outperform conventional state-of-the-art methods on these tasks. On the SUN397 scene classification, a performance of 40.95% was reported as opposed to the then-best 38% by [139], even though the deep features were trained on images depicting object categories. In a similar study, [107] demonstrated that deep features as learned by yet another existing, contemporary CNN architecture can be employed as off-the-shelf descriptors for a variety of vision tasks including attribute detection and fine-grained

recognition. A 69% classification performance on the MIT Indoor67 was reported, easily outperforming the then-state-of-the art methods. In similar vein, [153] then demonstrated state-of-the-art MIT Indoor67 recognition performance by combining DeCAF with SIFT, or with features learned based on their own DSFL learning framework (which produces discriminative features that are also shareable across classes).

Other recent works have sought to extract CNN descriptors on a multi-scale image representation, followed by encoding into an image-level descriptor. [41] employ of Bag-of-words like orderless encoding of deep features extracted at multiple spatial regions and scales, demonstrating state-of-the-art rates on both the MIT Indoor67 as well as SUN397 scene classification tasks. [15] also employ pre-trained features, but instead of using the output of a fully connected layer, they perform Fisher Vector pooling on the output of the last convolutional layer, reporting the best to-date performance of 81% on the MIT Indoor67 (see Tables 2.1, 2.2).

2.10 Benchmark Datasets for Scene Recognition

Scene category recognition is a relatively recent area of research in contrast to face, object or texture classification. As such, large scale benchmark datasets for scene classification have only been made available in the recent past. An early benchmark dataset for 8 outdoor scene categories was introduced by [94], and contained a mix of urban categories (e.g., highway, tall buildings, etc) and natural landscape (mountains, coast, forest, etc). An additional outdoor (suburb), along with 4 indoor categories (bedroom, kitchen, livingroom and office), were added to this set by [32].

Two more categories, including one indoor (store), were included by [70] to make for a 15-category dataset that has since widely appeared in literature [141, 62, 146, 134, 34].

The seminal work of [106] presented the then-largest testbench of 67 **indoor** scene categories called the MIT Indoor67 dataset. It contains a total of 100 images per category, with around 80 for training and the rest for testing. This dataset posed a substantial increase in difficulty over the earlier 15-category dataset, since an algorithm now not only needs to be scalable to the large number of categories, but also deal with the significant within-class variation manifested in indoor images. This dataset therefore firmly established indoor scene recognition as an open research problem in computer vision. *All* indoor scene images appearing in this thesis are taken form this dataset.

An even larger SUN (Scene UNderstanding) database sporting 899 scene categories and 130,519 images has since been made available [139, 138]. For scene classification, the benchmark specifies a subset of 397 well-sampled categories (those containing at least 100 unique images). However, this dataset is a mix of indoor and outdoor categories.

Following the ImageNet object image repository, a large scale dataset of 8+ million images depicting 401 unique scene categories (both indoor and outdoor) has appeared recently, called Places2 [150] (evolving from a former Places dataset [151]). For each category, it contains between 4,020 to 30,000 training images, 50 validation images (with labels) and 950 test images (whose labels are not available to the public). Consequently, this dataset has facilitated a large scale scene classification track at the ILSVRC [113] since 2015.

2.11 State of the Art in Indoor Scene Recognition

Most works appearing in literature over the years addressing the MIT Indoor67 dataset have been touched upon in the review conducted in this chapter. In addition, Tables 2.1 and 2.2 compile the most recent results on this dataset, that essentially define the state of the art in indoor scene recognition. Chapter 6 presents the classification results arrived at by this thesis on this challenging dataset. Moreover, qualitative results for rectification and detection of homogeneous texture in this dataset are presented at various points in Chapters 4 and 5.

Single Rep.	**% Accuracy**
(OPM) (CVPR'14) [140]	51.45%
Mode Seeking (NIPS'13) [23]	**64.03%**
SIFT (CVPR'13) [63]	60.77%
BoP (CVPR'13) [63]	46.10%
DSFL (ECCV'14) [153]	52.24%
DeCAF (CNN: AlexNet) (ICML'14 [25]) [153]	58.52%
MOP-CNN (CNN: Caffe) (ECCV'14) [41]	68.88%
CNN-SVM(CNN: OverFeat) (CVPRW'14) [107]	58.4%
CNNaug-SVM(CNN: OverFeat) (CVPRW'14) [107]	69.0%
FC-CNN(CNN: VGG-M) (CVPR'15) [15]	67.6%
FV-CNN(CNN: VGG-M) (CVPR'15) [15]	**81**%

TABLE 2.1: MIT Indoor67 classification — state of the art (single representation). All methods (except SIFT) employ learning based feature extraction. For a fair comparison, note that methods in the bottom half employ deep features pre-trained on the massive ILSVRC dataset [113] as off-the-shelf descriptors

.

Combined Rep.	**% Accuracy**
BoP + SIFT (CVPR'13) [63]	63.10%
OPM + SPM (CVPR'14) [140]	63.48%
Mode Seeking + SIFT (NIPS'13) [23]	66.87%
ISPR + SIFT (CVPR'14) [81]	**68.5%**
SIFT + DeCAF (ECCV'14) [153]	70.51%
DSFL + DeCAF (ECCV'14) [153]	**76.23**%

TABLE 2.2: MIT Indoor67 classification — state of the art (combined representation). All methods (except SIFT) employ learning based feature extraction. For a fair comparison, note that methods in the bottom half employ deep features pre-trained on the massive ILSVRC dataset [113] as off-the-shelf descriptors

.

Chapter 3

Indoor Scene Recognition: Possibilities & Challenges

In this chapter, a discourse is presented, articulating possible approaches to indoor scene recognition, and identifying the hurdles in practically realizing them. Sec. 3.1 picks up from Sec. 2.6, which surveyed the promising line of work on discovering mid-level features for indoor scene recognition. The pros of such an approach are reiterated, but the challenges are also detailed. Sec. 3.2 muses over the possibility of exploiting indoor scene geometry for classification, identifies existing work that may be leveraged to do so, but goes on to find it is not ready to be put to such use in current form. Sec. 3.3 provides a brief overview as to how the remainder of this thesis proposes to address the challenges brought out in the current chapter.

3.1 Top-Down Recognition via Mid-Level Features

Low-level patch-based features (see Sec. 2.3.1), though allowing local positional invariance, lack the context and semantics needed to reliably generalize to perceptually similar, or discriminate between differing image regions [65]. Recent sophisticated encoding schemes (Sec. 2.3.2, [11, 54]), however, have demonstrated impressive classification performance with low-level features employed in a bottom-up dictionary-based pipeline, especially when coarse spatial information is also preserved ([95]).

On the other hand, exploiting semantically meaningful image regions or parts has great potential, especially for indoor scenes (Sec. 2.6). These "mid-level" visual features possess the spatial support necessary to generalize well to similar instances in the face of intra-class variation (see Fig. 2.4). Additionally, such a top-down, object-detection style approach is potentially amenable to a more principled spatial constraint and contextual modeling [22, 12, 56], that can improve detections and minimize false alarms. Unfortunately, a practical pipeline implementing a top-down, mid-level feature based approach to scene recognition is not easy to realize. In what follows, some of the challenges in this direction are identified.

3.1.1 Image Annotation: Cumbersome, Expensive and Error-Prone

With the recent availability of large scale datasets sporting hundreds of categories and tens of thousands of images [139, 21], human annotation becomes increasingly challenging, costly [98, 97], and susceptible to error.

Annotation may have been performed with one task in mind, while it may later be desired to employ the dataset for some other task. Moreover, it is sometimes desired to use an out-of-the-blue, custom dataset for a specialized task [24], and this dataset might not be annotated. Finally, generalization of a model learned on one dataset to another is not always guaranteed (as evidenced by the low performance of a top-down indoor scene recognition system using pre-trained object models [78], compared to models learned automatically from the target dataset [63, 23]). Therefore, it becomes increasingly necessary to invest in research to minimize the level of supervision.[1]

3.1.2 Automatic Discovery of Mid-Level Features: A Chicken-and-Egg Problem

Give the modern-day availability of powerful and parallel computing resources, coupled with advanced machine learning algorithms (see [119]), unsupervised learning ([24]) seems to be an exciting direction. However, such an approach is not trivial, as discussed below.

Ill-posed: The approach is inherently ill-posed in that neither the appearance models of the patches sought are known (hence, one cannot detect them in a given image), nor are their occurrences in given images (hence, one cannot train detectors for them). Even with supervision, the trained HOG detectors are susceptible to raising false alarms [131], and, given the large presence of clutter and intra-class variation of all nature in real

[1] This is not to say that this thesis advocates an un-supervised, or a supervised for that matter, learning based approach — it does neither, nor does it altogether reject them. It merely identifies the potential problems in these directions, and goes on to propose a non-learning based method to detect meaningful, mid-level features in Chapter 5.

indoor images, the task is rather difficult. To appease the ill-posed nature of the problem, a common constraint employed by all (to best of the author's knowledge) proposed algorithms in this direction is that of discriminativeness — part models are learned such that they fire strongly on images of the concerned scene category versus those of all other categories ([95, 119, 63, 23], see Sec. 2.6). However, it is arguable whether enforcing such *scene* discriminativeness at the outset necessarily leads to good *part* models.

Viewpoint differences: Indoor scenes are photographed from differing viewpoints, and gradient based HOG (Sec. 2.3.1), the de-facto standard features in object detection, are *not* invariant to viewpoint differences. Consequently, two semantically similar parts differing somewhat in viewpoint would be treated as different parts, modeled by two different part models. This would be fine, except for the fact that we may not have a sufficiently huge dataset available, depicting *all* parts in *all* viewpoints to learn appearance models from.

Occlusions: Where mid-level features are better at the task of generalization compared to low-level features, the opposite is true when it comes to occlusion handling. Very often, an object or part will be partially or fully occluded in a given scene. Already a bane in supervised object detection [135, 99], occlusions, widely manifested, exacerbate the task of automatic learning. A partial part detection, if admitted during an iterative learning process, such as [119], can adversely affect the evolution of the part model. In the course of experimentation for this thesis, unsupervised learning of tree models for indoor scenes (see [152], a supervised method for face detection and pose estimation, and [13], which models hierarchical context for objects), jointly with part models, was unsuccessfully attempted in order

to impose contextual constraints on the part discovery process. Part occlusions were found to be one major factor for failure, since a single occluded node in the tree hierarchy can severely impact the dynamic programming based inference process. Should a joint discovery of scene parts be pursued in future (as a potential constraint to the part learning process in addition to discriminative learning), it would perhaps be more pertinent to instead model local ensembles of objects (see [76]) instead of a global, occlusion-prone, tree modeling.

3.2 Exploiting Indoor Scene Geometry

The geometry of an indoor scene is highly constrained. Not only do indoor scenes exhibit a predominantly planar structure, but most of the manifested planes also tend to be aligned along a few principal directions. We humans can effortlessly, yet accurately, discern not only the major room planar surfaces — ceiling, walls and floor — but also any 'secondary' surfaces, i.e., horizontal and vertical planes making up the contained furniture, such as the top of a table or bed, or the frontal view of a bookcase. Additionally, we can do so from any viewpoint of a given scene, or in the presence of unwanted interference such as room clutter or photometric severities (insufficient illumination, change in lighting conditions across a given scene).

If a machine could be equipped with such high-level vision capabilities, the resulting, semantically meaningful, scene segmentations could potentially be exploited to influence recognition. This is so since the various room surfaces exhibit characteristic properties that are unique (in classification jargon, "discriminative") to the design and decor of a given scene category. For instance, walls in a kitchen scene are typically lined with cabinets above and counters below. The floor in a classroom is covered with rows of

chairs and desks. The ceiling in a living room can often sport a chandelier. Secondly, provided the vanishing points of a segmented scene plane can be estimated, a planar rectification may be performed to restore the viewpoint to a 'canonical' form. Features extracted upon such a rectification are likely to better match with those from a similar scene region in another image of the same scene, possibly depicting a somewhat differing viewpoint.

Unfortunately, as hard as it is to computationally perform a fine-grained generic multi-object segmentation in images, the coarse-grained room segmentation we seek is equally difficult! The following sub-section reviews some existing work along this direction, and identifies their shortcomings. Furthermore, this thesis focuses on recovering planar scene structure from *single* images, and therefore does not explore approaches exploiting depth sensors (e.g., [132]), or those based on multiple views of a given scene such as stereo and motion (e.g., [126]).

3.2.1 Automatic Estimation of Spatial Layout: Issues in Real-World Images

Hoiem et. al. [52] have previously demonstrated the use of a rich set of color, texture, shape and geometric local features to learn appearance based models for the geometric classes of scene regions. Three main classes are defined: ground, vertical planes and sky. The vertical planes are further divided into three planar (left-facing, frontal and right-facing), and two non-planar (porous and solid) subclasses. A learned pairwise affinity function is used to obtain multiple hypotheses of scene region segmentations by grouping image superpixels into scene regions. Boosted decision tree classifiers are used to obtain the likelihood of whether all superpixels in a given scene region have the same geometric class label, and that of

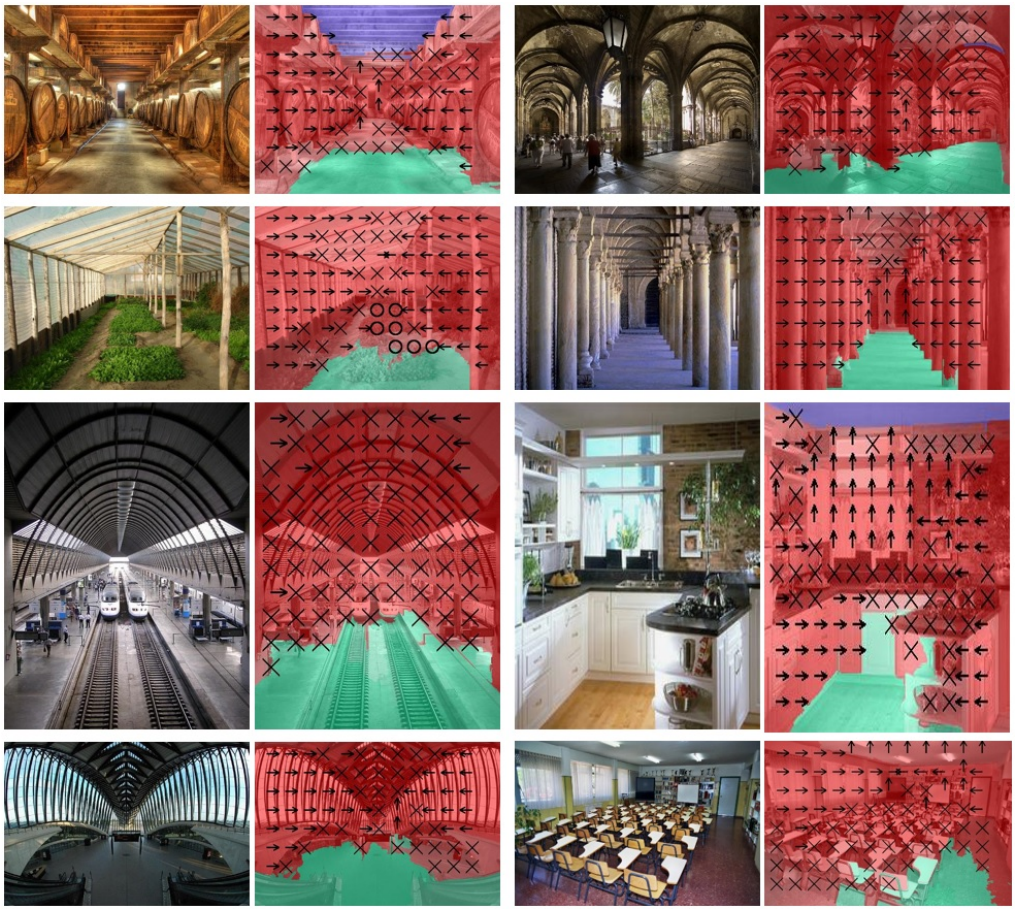


FIGURE 3.1: Illustration of the scene geometric context estimation method of [52], and sample results for the MIT Indoor67 dataset. Main geometric classes are shown by colored overlay (vertical = red, ground = green, ceiling/sky = blue). Subclasses are shown by markings (left, up, right arrows indicate planar surfaces; "X" and "O" indicate solid and porous surfaces, respectively). **Best viewed in color.**

the region label. Finally, a superpixel's label confidence is computed as a weighted average of region likelihoods.

Initially proposed for natural, suburban and urban scenes, [52] also re-trained the classifiers on indoor images facilitating a significant improvement in geometric class labeling for indoor scenes. However, they only demonstrate the approach on simple corridor-like scenes with no or little room clutter. In experiments for this thesis, the method was observed to

not fare so well on samples from the MIT Indoor67. Fig. 3.1 illustrates some successes and failures obtained using author implementation of the approach. We observe that the left, right and frontal vertical surfaces are quite often accurately labeled. Plants are correctly labeled as porous (2nd row, 2nd column). However, more often than not, many surfaces are misclassified. While the sky/ceiling is fully (3rd row, 4th column) or partially (1st row, 2nd column) recovered in a few cases, it is mostly misclassified as a vertical surface. The ground is also often partially misclassified as vertical surface, especially in the presence of clutter (2nd row, 2nd column and 4th row, 4th column). Seemingly, the method also cannot, in general, be relied upon to recover fine-grained horizontal surfaces such as table tops.

During the preliminary phase of this thesis, an MIT Indoor67 classification experiment was performed based on the scene geometry recovered by this approach. Essentially, we would like to see whether the estimated coarse scene structure provides a better alternative to the fixed grid based scene partitioning popularized by the spatial pyramid scheme (see Sec. 2.3.2, [70]), which assumes the scenes are roughly aligned in space. Specifically, SIFT features are pooled over five spatial bins: the entire image, ceiling, floor, vertical surfaces (including all its five sub-classes), and a fifth bin containing only the image regions classified as solid or porous (the conjecture being that these characterize the room objects or clutter). To obtain scene segmentations, pre-trained classifiers as provided by the authors and their own software implementation was employed. This is compared to a usual two-level spatial pyramid representation (also containing five bins). Fisher Encoding is used with one-vs-all SVMs. The details of the parameters and configuration may be looked up in Sec. 6.1. Table 3.1 presents the results. While a reasonable classification performance of 57.21% is attained, it is surprising to see a fixed spatial grid based representation outperforming a

Method	**% Accuracy**
FE_SIFT (spatial pyramid)	59.14%
FE_SIFT (geometric context)	57.21%

TABLE 3.1: MIT Indoor67 classification performance with Fisher-encoded SIFT — 2-level spatial pyramid representation vs. binning based on scene regions recovered by Geometric Context [52].

representation based on a more principled scene segmentation. Indeed it is unreasonable to expect the method of [52] to take on all the various kind of scenes in this challenging dataset. It might pay to perhaps re-train the classifiers on this dataset for improved generalizability. However, such an approach entails a lengthy and cumbersome annotation process. Additionally, this method in its original form does not model any plane projective parameters that can be used to perform a planar rectification — a central theme in this thesis — in order to push recognition performance.

Representative approaches that can deliver said planar scene structure from single images include [51, 121], and rely upon estimating scene vanishing points, modeling the room via a box layout as if it were empty. Let us analyze Hedau et. al. [51] in some detail. Connected component analysis is used to obtain long, straight lines, that are then clustered into three mutually orthogonal directions by imposing certain orthogonality criteria [112]. The point of intersection — essentially, the vanishing point — for each cluster may be computed by a voting based scheme, as in [51], or via linear least squares, optionally obtaining robust maximum-likelihood estimates by minimizing errors in the estimation of lines [47]. This fixes the room box orientation. The remaining problem now is that of obtaining the exact translation and scale of a given room face (walls, ceiling, floor). [51] propose to sample a set of rays emanating from each viewpoint, the intersection of which yields a set of candidate box layouts [Fig. 3.2 (left)]. They

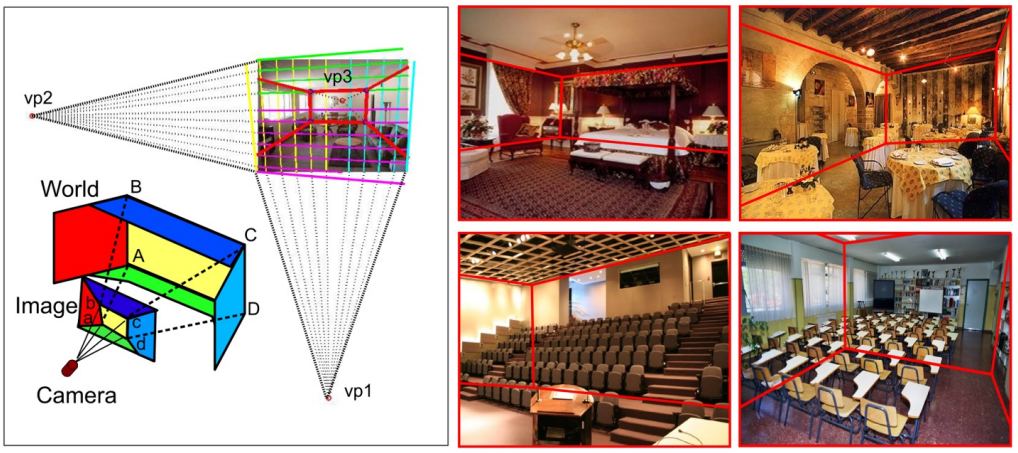


FIGURE 3.2: Illustration of the room spatial box-layout estimation method of [51] (left; taken from [51]), and sample results for the MIT Indoor67 dataset (right).

then learn structured SVMs (using a set of training images with annotated layouts) to rank the candidate room layouts using edge-based features. The method of [52] is modified to also make use of vanishing points and features based on the box layouts to obtain scene segmentations with geometric labels. These surface labels, and features computed over them, are then used to re-rank the box layouts, finally proposing the best scene layout. The conjecture is that a joint modeling of a coarse box layout and scene surface labels can improve the estimation performance for both, especially reducing the effects of clutter (room content) in box layout estimation. Some sample layouts are depicted in Fig. 3.2 (right). An impressive resilience to clutter (bed, auditorium seats, dining tables and classroom chairs) can be seen.

Indeed, the "Manhattan" structure [17] is well manifested in indoor scenes where surfaces are planar and aligned along three mutually orthogonal directions, and therefore such an approach seems attractive. However, in experiments for this thesis, it was observed that these simplistic assumptions are often violated in real images, among other challenges. Fig. 3.3

applies the method of [51] to some typical images in the MIT Indoor67, using author implementation. One observes the following:

Incorrect room face localization: Even in scenes where vanishing points may be reliably estimated [Figs. 3.3(a), 3.2(classroom)], localization of the faces in space and scale (albeit the heavy use of machine learning) is not always possible. As such, characteristic features in one room face can end up being assigned to the wrong room face.

Inability to handle forked layouts: Fig. 3.3 (b) depicts a forked scene layout that violates the box assumption — though arguably Manhattan — and therefore cannot be properly handled.

More than three dominant planar directions: The scene in Fig. 3.3 (c) features two additional planar directions due to an angled ceiling, besides the usual three. Imposing orthogonality [112] to recover vanishing points in such a scenario understandably fails. In the course of experimentation for this thesis, a greedy voting based strategy to compute vanishing points was implemented. Such a presence of more than three principal directions in indoor scenes — widely manifested in practice — was observed to be a major failure cause for estimating vanishing points (which is already an ill-posed problem), besides clutter.

Non-existent straight lines in a principal direction: Fig. 3.3 (d) shows a row of columns, suggesting a vertical planar structure that is tilted away from the camera. While the scene seemingly satisfies the box layout, there are no straight lines in the direction along the camera principal axis. Thus the corresponding vanishing point cannot be obtained, adversely affecting the room layout estimate.

Non-Manhattan indoor structure: A broad category of scenes in fact *do not* conform to a Manhattan structure wherein surfaces are strictly

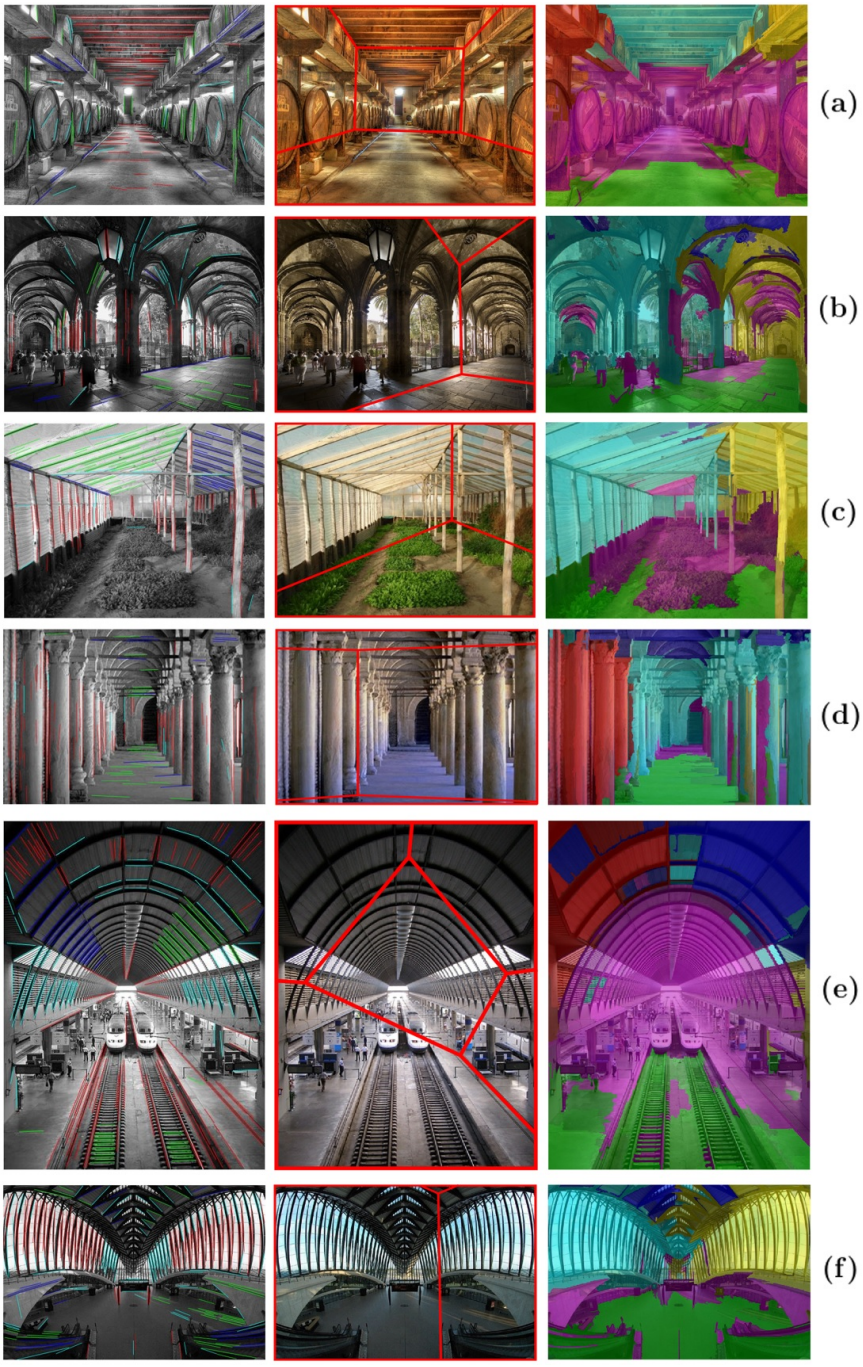


FIGURE 3.3: Failure cases of the spatial layout estimation method of [51]. Each row depicts, in order, line clusters assigned to vanishing points, box layout, scene geometric labels ([51]+[52]) (a) incorrect room face localization; (b) inability to handle forked layout; (c, d) incorrect vanishing point(s), and hence viewpoint, estimates due to lack of straight lines in a principal direction, or due to manifestation of more than 3 dominant planar directions; (e, f) not applicable to a broad category of scenes that don't conform to a conventional box layout. Left wall = red, mid wall = cyan, right wall = yellow, ceiling = blue, floor = green. See also Fig. 5.5. **Best viewed in color.**

planar, and aligned only in three directions. Figs. 3.3 (e, f) depict scenes from a train station and an airport as common cases.

3.3 The Way Forward

The remainder of this thesis attempts to address some of the problems highlighted in this chapter in order to facilitate indoor scene recognition. In this regard, instead of relying upon the obvious but restrictive (as demonstrated in this chapter) Manhattan scene assumption, Chapter 4 will exploit a more general assumption in indoor scenes — that they strongly exhibit regularly repeating planar structure, or in other words, homogeneous texture — and propose robust methods to recover projective parameters from such structure. In doing so, reliance upon straight lines is no longer required, and vanishing points need not be computed, though when available can be leveraged upon (see Sec. 5.4). Chapter 5 will consequently show that machine learning need not be invoked to localize planes in space and scale in real world indoor scenes, provided they satisfy homogeneity. Any homogeneous room content is not treated as "clutter", but also localized, providing a more fine-grained modeling of room layout, as opposed to [51]. At the same time, this provides for a non-learning based approach to detecting meaningful mid-level features, useful for scene recognition, as shall be demonstrated via a comprehensive set of experiments in Chapter 6.

Chapter 4

Affine Rectification of Planar Homogeneous Texture

Sec. 4.1 draws attention to the abundance of homogeneous texture in indoor scenes, motivates the planar rectification of such texture to improve scene recognition performance, and underscores the challenges faced by existing schemes for rectification. Sec. 4.2 reviews related work, contrasting it with the method proposed herein. Sec. 4.3 develops the texture frequency projection model that can be used for planar affine rectification. Sec. 4.4 analyses an existing approach to estimate dominant frequency in given texture, identifies and addresses two of its short-comings. Sec. 4.5 employs robust estimation to recover projective parameters, while Sec. 4.6 demonstrates an anisotropic multi-scale representation to further improve performance. Finally, Sec. 4.7 presents comprehensive qualitataive and quantitative results, demonstrating superior performance of the proposed scheme over existing work on some challenging texture from real-world indoor scenes.

4.1 Motivation

Indoor scenes tend to be abundant with planar structure. Besides the main architectural surfaces — ceilings, walls and floors — the content of an indoor scene, such as furniture, cabinets and countertops is all planar. Furthermore, there is a strong presence of regularly repeating structure or motifs, aligned along planes. Consider the examples in Fig. 4.1, appearing in the MIT Indoor67 dataset. One observes aligned columns in cloisters or corridors, rows of pews in churches, repeating steps on a staircase, etc. Next, Fig. 4.2 depicts patterned tiling, brickwork, wooden flooring and printed carpeting (also from the MIT Indoor67). Again, such kind of uniform patterns are all very characteristic of man-made indoor scenes, and, additionally, occur as planes. The aim in this chapter is to perform projective rectification on such kind of planar structure found so abundantly in indoor scenes. The next chapter demonstrates the detection of such patches in indoor images, and uses them in turn for scene classification.

In the absence of motion or stereo, shape-from-texture may be employed for said rectification. In this thesis, the term "texture" is used to refer to both the former, *architectural* or *structural* patterns, as well as the latter, more *conventional* patterns appearing, for e.g., on tiles or fabric. We invoke the notion of homogeneity in shape-from-texture, which requires that density and scale of texels be uniform across the plane. This assumption sits well with the kind of patterns we have just observed. Any deviation in homogeneity may then be attributed to perspective projection, and exploited to recover plane normal or transformation.

A recognition system can benefit from planar rectification as it mitigates in-class variation due to differences in viewpoint. The top row in each set of 2x3 patches in Fig. 4.1 depicts a triplet of patches from similar indoor

FIGURE 4.1: Indoor scenes are abundant with planar homogeneous texture. Rectification of such texture can reduce intra-class variation due to viewpoint differences. All depicted texture was detected (see Chapter 5) and rectified automatically via the proposed approach.

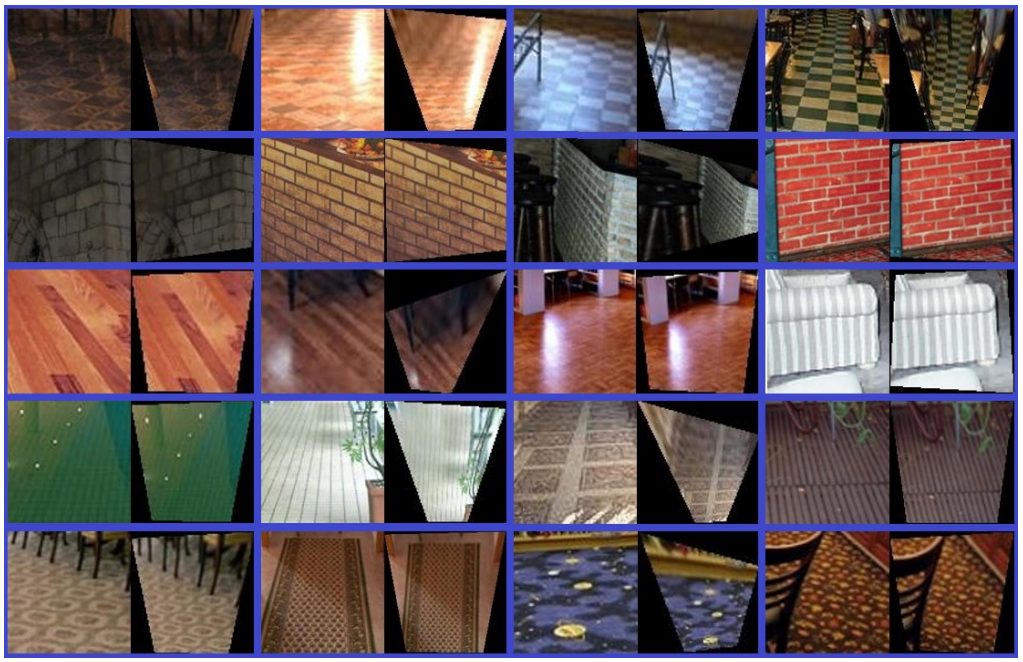


FIGURE 4.2: Examples of the more conventional texture manifesting in indoor scenes. All depicted texture was detected (see Chapter 5) and rectified automatically via the proposed approach.

scenes, but with significant viewpoint differences within the triplet (among other kinds of intra-class variation). Gradient based image descriptors typically employed in recognition, such as SIFT or HOG, are not invariant to perspective transforms, hence this can limit the performance of a recognition system. Upon planar rectification (bottom row in each set of 2x3 patches), it can be seen that patch gradients align along a canonical coordinate frame. A limitation of the proposed approach is that only the projective parameters of the homography are recovered, while any accompanying affine transform is not. This means that any rotation or anisotropic scale on the plane is not recovered, and the rectification contains an affine ambiguity. Nevertheless, it shall be observed in Chapter 5 that even affine-ambiguous rectification goes on to yield class-discriminative features that help improve recognition performance (it should be noted that the *detection*

in Chapter 5 is invariant to projective transforms, *including* any accompanying affine transforms).

Rectification of such real-world images is not straight-forward, however. The challenges include the presence of outliers (openings in courtyard columns, regular shelf pattern interspersed with irregular grocery items or books, see Fig. 4.1), illumination changes and shading, severely marring otherwise uniform patterns on the floor (Fig. 4.2), and clutter (indoor scene objects and furniture irrelevant to the pattern of interest, e.g., tables in front of laundry machines in Fig. 4.1, or chair legs on patterned flooring). In addition, the limited span or support of the texture in the patch (laundromat, wine barrels, bookshelves in Fig. 4.1) poses problems. Existing approaches to texture rectification have usually been demonstrated on cropped texture and sparse noise (see Sec. 4.2), while our application-oriented setting is significantly more challenging. This chapter, therefore, mainly aims to address these problems in planar rectification, and Sec. 4.7 shall compare the proposed approach with existing work in light of said challenges.

4.2 Related Work

Planar rectification is a well-studied problem. In an early work, [80] adopted a stratified approach where an affine rectification is obtained by first recovering the vanishing points, and hence the vanishing line. The rectification is then upgraded to a similarity assuming known metric properties in the world plane (they also show that direct rectification is possible from metric information). However, the method is not applicable in our setting as we do not have such prior knowledge available, and additionally we deal with multi-planar scenes. Approaches exist in literature that attempt to automatically detect dominant rectangular planar structure in simple,

non-cluttered indoor or urban environments [117, 67], or that detect primary indoor faces (walls, ceiling, floor) by employing sophisticated machine learning [51], or detect depth-ordered planes [121]. However, *all* these approaches assume the scene is aligned with a triplet of principal directions defining the coordinate frame, and that these directions can be reliably recovered in a scene. It has already been discussed in Sec. 3.2.1 that both these assumptions are often not valid for practical real-world indoor scenes.

The work presented here taps into classical shape-from-texture (SFT) theory — in particular the class of methods that work with planar homogeneous texture [111, 123]. However, unlike SFT, our goal here is not to recover surface normal but to perform planar rectification. We therefore reparameterize the local change in dominant texture frequency [123, 122, 49] as a function of the plane projective homography instead of the surface slant and tilt. The resulting formulation circumvents the need to define and relate coordinate systems and, more importantly, does not require knowledge of focal length, hence has wider applicability. One notes that [16] have previously presented a SFT system that does not require a calibrated camera, and jointly recovers surface normal and focal length. However, the system only works in the limited scenario where the fronto-parallel appearance of the texture is known a priori. On the other hand, as motivated in Sec. 4.1, we only make the weak assumption of texture homogeneity.

Criminsi and Zisserman [18] have also previously demonstrated recovering vanishing lines from projected homogeneous texture by exploiting the observation that the direction of perspective gradient is orthogonal to the vanishing line. However, the approach involves a computationally expensive search for the direction of maximum variance of a similarity measure, seems to be susceptible to such parameters as the size of image patch to compute the measure over, and has only been demonstrated on cropped

texture exhibiting a grid structure. On the other hand, [108] observed lines of equal spectral power are perpendicular to the perspective gradient, but they recover tilt and not a homography.

Similar to [80, 14], the approach presented in this chapter models and recovers the projective part of the homography. However, whereas [14] exploit relative scale change in recurring instances of affine-covariant MSER features (see Sec. 2.3.1), Sec. 4.3 exploits the local change in dominant texture frequency to obtain an affine rectification. A frequency based approach [123], as opposed to one involving feature detection [14, 104, 3], is capable of describing any generic homogeneous texture, and not necessarily composed of texture elements (texels) that can be sensed by a given feature detector (lines, blobs, edges, etc). Furthermore, as is demonstrated in Sec. 4.4, employing a frequency based texture representation allows us to make use of energy minimization methods to robustly track a dominant texture frequency component in the presence of outliers with large spatial support. Combined with robust parameter estimation (see Sec. 4.5), we arrive at a powerful approach that performs well in the face of aforementioned limited support and clutter. While the TILT algorithm of [149] directly employs raw pixel values, and does not involve low-level feature detection, it is applicable to a limited class of texture — that which upon rectification gives a low-rank matrix. Therefore, the approach has been successfully demonstrated only on a limited type of images — mainly faces, text and building facades. Furthermore, a region of interest often needs to be specified for the approach to work well. Moreover, the algorithm is explicitly designed to cater to spatially sparse noise, and hence may not work well with the outliers or clutter encountered in real-world scenes and mentioned in Sec. 4.1. Similarly, [3] demonstrate a resilience to sparse, salt-and-pepper like

noise. This assumption of sparse noise is hardly valid for texture in real-world indoor scenes, which, on the contrary, can often contain large blobs and blotches of outliers! In Sec. 4.7.1, qualitative comparisons are provided between the proposed method and those in [149, 3] on challenging real-world test cases that exhibit limited spatial support, large clutter and illuminaion changes.

In [104], an upgrade to the affine rectification of [14] within a similarity of the scene plane is demonstrated by making use of recurring rotated instances of a motif, provided they can be detected and matched intra-image. It shall be demonstrated later in Sec. 5.4 that if the scene vanishing points are known, they can be used in conjunction with estimated instantaneous frequency to automatically assign the correct pair of vanishing points to a region of homogeneous texture, and simultaneously obtain a rectification up to only a scale ambiguity for such regions.

4.3 Texture Frequency Projection Model

One class of shape-from-texture algorithms assumes an *isotropic* surface texture, i.e., it has no dominant orientation or bias (see [111, 40]). The deviation in isotropy upon (either orthographic or perspective) projection is used as a cue to recovering shape (e.g., a circle projects to an ellipse).

Another class of algorithms makes a more general assumption involving some form of texture *homogeneity* [111, 123, 124, 69, 18, 14, 104, 3]. When projected to the image plane, *texture gradients* come into play that cause the texture to deviate from homogeneity. The scale, area or perspective gradient is the shrinking of a texel as it recedes from the camera, the density gradient is the increased crowding of texels as they move farther, whereas

the compression gradient or foreshortening compresses a *given* texel along the direction of slant more than in the direction orthogonal to it. While all three are manifested in perspective projection, only foreshortening is present in orthographic (affine) projection. The approach in [14, 104], for e.g., exploits the scale gradient, while the density gradient is not explicitly modeled. Since an affine-ambiguous homography is recovered, the compression gradient is also not recovered.

The model developed in this section assumes texture homogeneity to imply that scale and density of texels (in the fronto-parallel view) is constant. A texture that has undergone an anisotropic scaling (due to some unknown affine transform) is still considered homogeneous. Hence, the compression gradient is not modeled. (Note the difference with [123, 122], who, by making use of an explicit perspective projection model given a calibrated camera, are able to implicitly exploit the compression gradient as well). The frequency domain equivalent of this assumption is that the texture should exhibit a *constant* spatial frequency content across the plane in a *given* direction. Sec. 4.4 demonstrates how to robustly track the instantaneous (point to point in spatial domain) dominant spatial frequency component in projected texture. In the current section, we shall attribute any local variation in spatial frequency — essentially, the deviation in texture homogeneity — to perspective projection; we then seek to undo this deviation in order to recover a rectifying homography up to an affine ambiguity.

Conventional shape-from-texture relates texture surface coordinates at a point to corresponding camera coordinates in terms of the slant and tilt of the tangent plane at that point [123, 124], or in terms of the plane gradients or normal [122, 69, 16]. Surface coordinates (expressed in camera reference frame) are then projected to the image plane via scaled orthographic or perspective projection. The transpose of Jacobian of the inverse of this

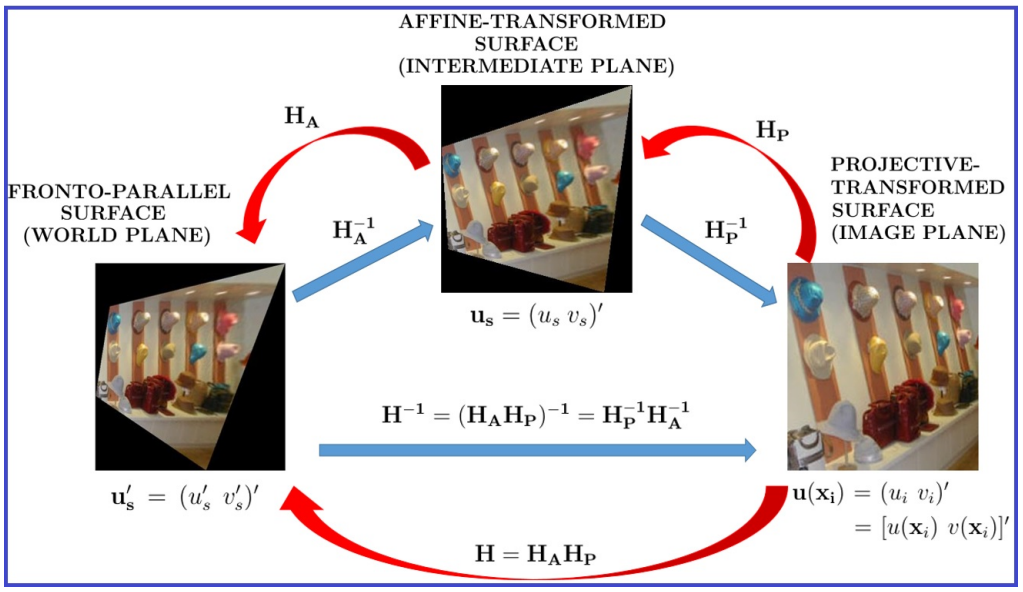


FIGURE 4.3: Texture surface projection — notations and geometry.

composite transformation (i.e., from image to surface coordinates) yields the transformation from surface to image spatial frequency [123]. Recovery of surface slant and tilt is not possible without knowledge of focal length, and all SFT systems assume this camera parameter is known. Since we are interested in planar rectification, we can relate the surface and image points via a planar homography instead of an explicit camera projection model. This does not require the focal length, but the downside, as we shall see shortly, is that we cannot recover any accompanying affine transform (i.e., rotation and anisotropic scale).

Fig. 4.3 depicts the projection geometry and the notations involved, using the example of an image from the MIT Indoor67 `clothingstore` category. The "texture" in this case is the pattern formed by the vertical hat hook bars. Observe that in the imaged plane (right image), the scale and density gradients discussed above are manifested, while compression gradient is not pronounced in this example. In the affine-rectified plane (top image), scale and density of texels becomes constant. Notice the limited support of the

pattern in the image, as well as the clutter (assorted hats at the bottom). Yet, the affine-rectified image was obtained automatically by the proposed approach. In the metric-rectified (fronto-parallel) plane (left image), any rotation and anisotropic scaling have also been removed (manually, since the proposed method does not support this).

Let us represent the projective transform from the image plane to the textured surface plane as a 3x3 homography H. This can be decomposed to separate the contributions of the affine part and the projective part [47]:

$$\begin{aligned} H &= H_A H_P \\ &= \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ h_7 & h_8 & 1 \end{pmatrix} \end{aligned} \tag{4.1}$$

In other words, the image coordinates are first transformed by the "purely" projective (i.e. what is left in the projective group after removing the affine group) homography to some intermediate plane, followed by the affine transform H_A to obtain the world (fronto-parallel) plane coordinates. We consider the role of H_A first. Let $\mathbf{x_s} = (x_s \ y_s)'$ denote the planar coordinates on said intermediate plane, which are transformed to world plane coordinates $\mathbf{x'_s} = (x'_s \ y'_s)'$ by H_A as:

$$x'_s = a_{11}x_s + a_{12}y_s + a_{13} \tag{4.2a}$$

$$y'_s = a_{21}x_s + a_{22}y_s + a_{23} \tag{4.2b}$$

The transpose of the Jacobian of H_A, given as:

$$\mathbf{J'_A} = \begin{pmatrix} \frac{\partial x'_s}{\partial x_s} & \frac{\partial y'_s}{\partial x_s} \\ \\ \frac{\partial x'_s}{\partial y_s} & \frac{\partial y'_s}{\partial y_s} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{21} \\ \\ a_{12} & a_{22} \end{pmatrix} \tag{4.3}$$

transforms a *given* world plane spatial frequency $\mathbf{u'_s} = (u'_s\ v'_s)'$ — which is constant over the entire plane, since we have assumed homogeneity of texture on the surface — into the frequency $\mathbf{u_s} = (u_s\ v_s)'$ on our intermediate plane:

$$\mathbf{u_s} = \mathbf{J'_A}\mathbf{u'_s} \tag{4.4}$$

i.e.,

$$u_s = a_{11}u'_s + a_{21}v'_s \tag{4.5a}$$

$$v_s = a_{12}u'_s + a_{22}v'_s \tag{4.5b}$$

Clearly, frequency $\mathbf{u_s}$ on the intermediate plane, albeit different from world plane frequency $\mathbf{u'_s}$, is also constant, i.e., does not vary spatially. In other words, homogeneous texture upon affine transform is still homogeneous, under our definition of homogeneity.

In a similar fashion, H_P transforms image points $\mathbf{x_i} = (x_i\ y_i)'$, into points $\mathbf{x_s} = (x_s\ y_s)'$ on our intermediate plane:

$$x_s = \frac{x_i}{h_7x_i + h_8y_i + 1} \tag{4.6a}$$

$$y_s = \frac{y_i}{h_7x_i + h_8y_i + 1} \tag{4.6b}$$

The transposed Jacobian matrix of the above function is:

$$\begin{aligned}\mathbf{J'_P} &= \begin{pmatrix} \frac{\partial x_s}{\partial x_i} & \frac{\partial y_s}{\partial x_i} \\ \\ \frac{\partial x_s}{\partial y_i} & \frac{\partial y_s}{\partial y_i} \end{pmatrix} \\ &= \frac{1}{(h_7x_i + h_8y_i + 1)^2} \begin{pmatrix} h_8y_i + 1 & -h_7y_i \\ -h_8x_i & h_7x_i + 1 \end{pmatrix}\end{aligned} \tag{4.7}$$

$\mathbf{J'_P}$ transforms the constant frequency $\mathbf{u_s} = (u_s \ v_s)'$ on the intermediate plane to variable frequency $\mathbf{u}(\mathbf{x_i}) = (u_i \ v_i)' = [u(\mathbf{x}_i) \ v(\mathbf{x}_i)]'$ on the image plane as:

$$\mathbf{u}(\mathbf{x_i}) = \mathbf{J'_P u_s} \tag{4.8}$$

While the above analysis is applicable to *any* spatial frequency component, in Sec. 4.4 we shall obtain a robust instantaneous estimate of the *dominant* spatial frequency component in a given image patch depicting real-world texture, which inevitably contains multiple frequency components. Denote said estimate as $\tilde{\mathbf{u}}(\mathbf{x_i}) = (\tilde{u}_i \ \tilde{v}_i)' = [\tilde{u}(\mathbf{x}_i) \ \tilde{v}(\mathbf{x}_i)]'$. We then arrive at a method to recover H_P by minimizing the following ***re-projection error*** over the projective parameters h_7, h_8 and the intermediate plane frequency u_s, v_s:

$$\begin{aligned} & E_{RP}(h_7, h_8, u_s, v_s) \\ & = \sum_{x_i} \sum_{y_i} \left(\frac{(h_8 y_i + 1)u_s - h_7 y_i v_s}{(h_7 x_i + h_8 y_i + 1)^2} - \tilde{u_i} \right)^2 \\ & + \sum_{x_i} \sum_{y_i} \left(\frac{(h_7 x_i + 1)v_s - h_8 x_i u_s}{(h_7 x_i + h_8 y_i + 1)^2} - \tilde{v_i} \right)^2 \end{aligned} \tag{4.9}$$

Eqn. 4.9 is an error measure in the image space. We may also define an error measure on the intermediate plane (where constant world-plane frequency is projected to another constant frequency via an affine transform) as follows. Consider H_P^{-1} that projects the intermediate plane to the image. The

corresponding transposed Jacobian:

$$\mathbf{J}'_{\mathbf{H_P^{-1}}} = \begin{pmatrix} \frac{\partial x_i}{\partial x_s} & \frac{\partial y_i}{\partial x_s} \\ \\ \frac{\partial x_i}{\partial y_s} & \frac{\partial y_i}{\partial y_s} \end{pmatrix} = \frac{1}{(1 - h_7 x_s - h_8 y_s)^2} \begin{pmatrix} 1 - h_8 y_s & h_7 y_s \\ h_8 x_s & 1 - h_7 x_s \end{pmatrix} \tag{4.10}$$

back-projects the variable image frequency u_i, v_i to constant intermediate plane frequency u_s, v_s. The ***back-projection error*** is then:

$$\begin{aligned} & E_{BP}(h_7, h_8, u_s, v_s) \\ & = \sum_{x_s} \sum_{y_s} \left(\frac{(1 - h_8 y_s)\tilde{u_i} + h_7 y_s \tilde{v_i}}{(1 - h_7 x_s - h_8 y_s)^2} - u_s \right)^2 \\ & + \sum_{x_s} \sum_{y_s} \left(\frac{(1 - h_7 x_s)\tilde{v_i} + h_8 x_s \tilde{u_i}}{(1 - h_7 x_s - h_8 y_s)^2} - v_s \right)^2 \end{aligned} \tag{4.11}$$

In Eqn. 4.11, back-projected coordinates $x_s = x_s(x_i)$ and $y_s = y_s(y_i)$ are obtained via Eqns. 4.6. Optimizing Eqn. 4.9 or Eqn. 4.11 is a nonlinear least squares problem, and may be performed via the Levenberg-Marquardt algorithm. The error measures indicate that parameters h_7 and h_8 reduce to 0 if and only if $u_i = u_s$ and $v_i = v_s$, respectively.

In experiments, the sum of the re-projection and back-projection errors:

$$E(h_7, h_8, u_s, v_s) = E_{RP} + E_{BP} \tag{4.12}$$

is minimized to yield more robust estimates for parameters h_7, h_8, u_s and v_s, rather than minimizing either error. Our rationale for combining the two error measures is as follows. One can view the operation performed by Eqns. 4.6 to obtain Eqn. 4.11 as some kind of data normalization, and

analogous to the case of estimating epipolar geometry [148], we empirically evaluate which data normalization yields the best results and arrive at Eqn. 4.12.[1]

Observe that our method allows the recovery of H_P and not H_A. This is because $\mathbf{J'_A}$ maps the fronto-parallel plane frequency $\mathbf{u'_s} = (u'_s \; v'_s)'$ to a different but still constant frequency $\mathbf{u_s} = (u_s \; v_s)'$. As such, a planar rectification only to within an ambiguous affine transform H_A^{-1} of the fronto-parallel plane may be obtained.

4.4 Robust Tracking of Dominant Frequency in Projected Homogeneus Texture

The 2D DFT captures the global spatial frequency content of the given image by specifying the magnitude and phase of each frequency (which ranges from 0 to 0.5 cycles/pixel, i.e., the Nyquist frequency). However, we are interested in estimating the spatially local (instantaneous) frequency content. This may be achieved with the Short-Term Fourier Transform (STFT), also called the windowed Fourier Transform. The STFT computes the local spectral content by applying DFT to small windows or patches in the given image. There is an associated trade-off between spatial and frequency domain resolutions. A compact spatial-domain window yields more local estimates in space, and vice versa. The special case where the window has a Gaussian form is called the Gabor transform, and has an optimal trade-off between time and frequency resolutions, i.e., maximum possible resolution in both domains simultaneously (see, e.g., [20]).

[1] For computational stability, the pixel coordinates are also normalized such that the top-left of the patch is given by (-1,-1) and the bottom right by (1,1).

Putting it mathematically, a Gabor filter:

$$h(\mathbf{u};\mathbf{x}) = \frac{1}{2\pi\gamma^2}\exp\left\{-\frac{\mathbf{x}.\mathbf{x}^2}{2\gamma^2}\right\}\exp\left\{2\pi j\mathbf{u}.\mathbf{x}\right\} \tag{4.13}$$

with effective width, receptive field, or standard deviation γ and spatial center frequency $\mathbf{u} = (u, v)$, can be convolved with an image $f(\mathbf{x})$, followed by evaluating the complex magnitude, to give its frequency content near $\mathbf{u}$ at spatial point $\mathbf{x} = (x, y)$:

$$A(\mathbf{u};\mathbf{x}) = |f(\mathbf{x}) * h(\mathbf{u};\mathbf{x})| \tag{4.14}$$

The above form of the Gabor function is as in [122, 123, 124]. It can be easily shown that it is equivalent to the parameterization proposed in [102, 116, 91] if the spatial aspect ratio of the filter is set to 1 (i.e., the filters have a circular rather than an elliptical shape).

Now, assuming texture homogeneity, we want to measure how a *given* frequency component (which would be constant over space sans projection) varies instantaneously (i.e., from pixel to pixel) in a certain direction so as to be able to use the projection model developed in Sec. 4.3. Moreover, since a given homogeneous texture may exhibit multiple frequencies, which may also be oriented differently, we must discern the component we can reliably track over space. In this regard, Super and Bovik [122, 123] have previously demonstrated estimation of the *dominant* texture frequency — a distinct peak at any given point, around which most of the energy is concentrated in a narrow band.

A naive approach to estimating the dominant frequency at a point in the image is to compute the responses at this point to Gabor filters with a dense sampling of center frequencies in the spatial frequency plane. The center frequency giving the maximum response is the required estimate.

However, a sufficiently dense sampling of center frequencies to provide an appreciably smooth instantaneous estimate is computationally infeasible. Super and Bovik have proposed more practical approaches involving combining estimates from multiple neighbouring filters in [122], or employing a frequency demodulation model from [49] for improved frequency estimates [123, 124]. In this section, the demodulation based approach (DEMOD) as presented by Super and Bovik is reviewed, and then applied to significantly more challenging texture compared to the original work in order to identify and address its shortcomings.

Let us denote the horizontal and vertical partial derivatives of Gabor filter $h(\mathbf{u};\mathbf{x})$ by $h_x(\mathbf{u};\mathbf{x})$ and $h_y(\mathbf{u};\mathbf{x})$ respectively, and the corresponding amplitude response (Eqn. 4.14) by $B(\mathbf{u};\mathbf{x})$ and $C(\mathbf{u};\mathbf{x})$ respectively. Then, an *unsigned* instantaneous estimate $|\tilde{\mathbf{u}}(\mathbf{x})|$[2] of a frequency component that lies in the passband of filter $h(\mathbf{u};\mathbf{x})$ is given by:

$$|\tilde{u}(\mathbf{x})| = \frac{B(\mathbf{u};\mathbf{x})}{2\pi A(\mathbf{u};\mathbf{x})} \tag{4.15a}$$

$$|\tilde{v}(\mathbf{x})| = \frac{C(\mathbf{u};\mathbf{x})}{2\pi A(\mathbf{u};\mathbf{x})} \tag{4.15b}$$

Equivalently, the associativity property of convolution $[f*(g*h) = (f*g)*h]$ may be invoked, and $B(\mathbf{u};\mathbf{x})$, $C(\mathbf{u};\mathbf{x})$ defined as the responses of the partial derivatives f_x, f_y of the texture image $f(\mathbf{x})$ to the Gabor $h(\mathbf{u};\mathbf{x})$. The *dominant* component estimate at each point $\tilde{\mathbf{u}}(\mathbf{x})$ may be computed by applying Eqns. 4.15 for the filter h that maximizes the response $A(\mathbf{u};\mathbf{x})$ at that point. Observe that only an unsigned estimate of the frequency is recovered. In their original work [123], the authors sample Gabor filters

[2]The symbol tilde (˜) is used to denote an *instantaneous* quantity in [123, 124]. In this thesis, however, it is used to denote an *estimated* quantity, while the *instantaneous* nature is already clear by writing it as a function of $\mathbf{x}$. As such, equality (=) is used in Eqns. 4.15 instead of the approximate equality (≈) appearing in [123, 124].

from quadrants I and IV of the frequency plane, and choose the quadrant of the maximizing Gabor at each pixel to define the signs of the horizontal and vertical frequency.

The Gabor filter bank used in all experiments for this thesis is described in the following, and differs somewhat from [122, 123, 124] since it was experimentally fine-tuned to our setting. Filters sized 45x45 pixels are generated via Eqn. 4.13. 6 radial center frequencies Ω are sampled along a geometric progression from 3 to 16.9706 cycles/image with a common ratio $\sqrt{2}$. As suggested in [123, 124], the bandwidth is fixed so that the effective width γ varies proportionally with the center frequency Ω. The proportionality constant may be computed as [102]:

$$\frac{\gamma}{\lambda} = \frac{1}{\pi}\sqrt{\frac{ln2}{2}}.\frac{2^b+1}{2^b-1} \tag{4.16}$$

where b is the half-magnitude response spatial bandwidth of the Gabor filter, set to 1 in all experiments. 10 radial orientations θ spanning quadrants IV and I are used, spaced uniformly by 18° i.e., $-90°$ to 72°. Finally, the relationship between the polar form (Ω, θ) and the cartesian form $\mathbf{u} = (u, v)$ of spatial frequency is defined as:

$$\mathbf{u} = (u, v) = (\Omega \sin\theta,\ \Omega \cos\theta) \tag{4.17}$$

The filter bank constructed above is illustrated in Fig. 4.4 by visualizing the real part of the complex-valued functions. The imaginary parts simply consist of a 90° offset relative to their real counterparts.

In its original form, the DEMOD approach reviewed above was found to perform rather poorly in our application setting of homogeneous texture in indoor scenes, which inherently exhibit clutter and outliers. The following

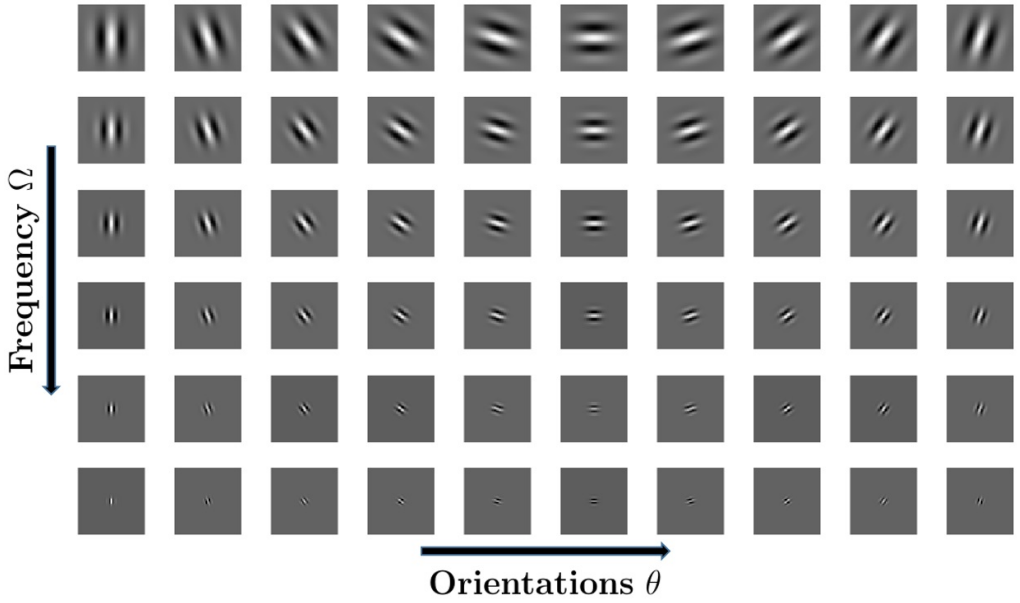


FIGURE 4.4: Visualization of the Gabor filter bank used in all experiments for this thesis. Only the real parts of the complex-valued functions are shown. The radial frequencies increase along a geometric progression from 3 to 16.9706, while the orientations are uniformly spaced from $-90°$ to $72°$ (see text for details).

sub-sections identify two shortcomings of DEMOD, namely frequency drift and quadrant ambiguity, and propose effective solutions.

4.4.1 Frequency Drift

Consider the 130x80 pixel patch in Fig. 4.5(a) depicting a glass ceiling cut out from an MIT Indoor67 `airport_inside` image. The texture in question is the lattice formed by the metal frame on the ceiling. As an aside, observe it is unreasonable to expect an algorithm that uses lines in the image to reliably compute the horizontal dominant vanishing point for this patch (notwithstanding the patch must first be segmented out in the image), since the horizontal bars are piecewise linear and not rectilinear.

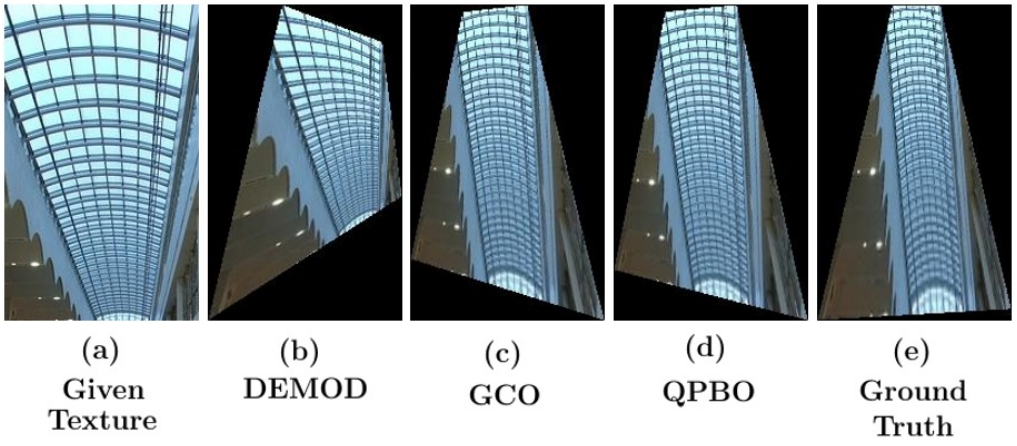


FIGURE 4.5: Affine rectification of given texture (a) via the model developed in Sec. 4.3 applied to dominant instantaneous frequency estimate. Non-optimal estimate via demodulation (b) is prone to drift, optimal estimates via GCO (c) or QPBO (d) improve performance. Ground truth is shown in (e).

The ground truth affine rectification, obtained by manual annotation of vanishing points, is shown in Fig. 4.5(e).

Estimating the dominant frequency in this image using the demodulation scheme just reviewed, and obtaining the projective parameters by minimizing Eqn. 4.12 results in a rather poor affine rectification (Fig. 4.5(b)).

The failure may be understood by inspecting the center frequency and orientation of the dominant Gabor filter (i.e., the one yielding the maximum response) at each pixel, as shown in Fig. 4.6(a) and (c) (brigher pixels depict numerically larger values). Since the given texture does not extend to the lower left and lower right regions in the image patch (Fig. 4.5(a)), the dominant Gabor estimate *drifts* in *both* the center frequency as well as the orientation in these regions. Fig. 4.6(b) and (d) plot the dominant center frequency and orientation, respectively, along the dotted lines in Fig. 4.6(a) and (c). The center frequency is seen to momentarily drop before continuing with its increasing pattern, and then dropping again. On the

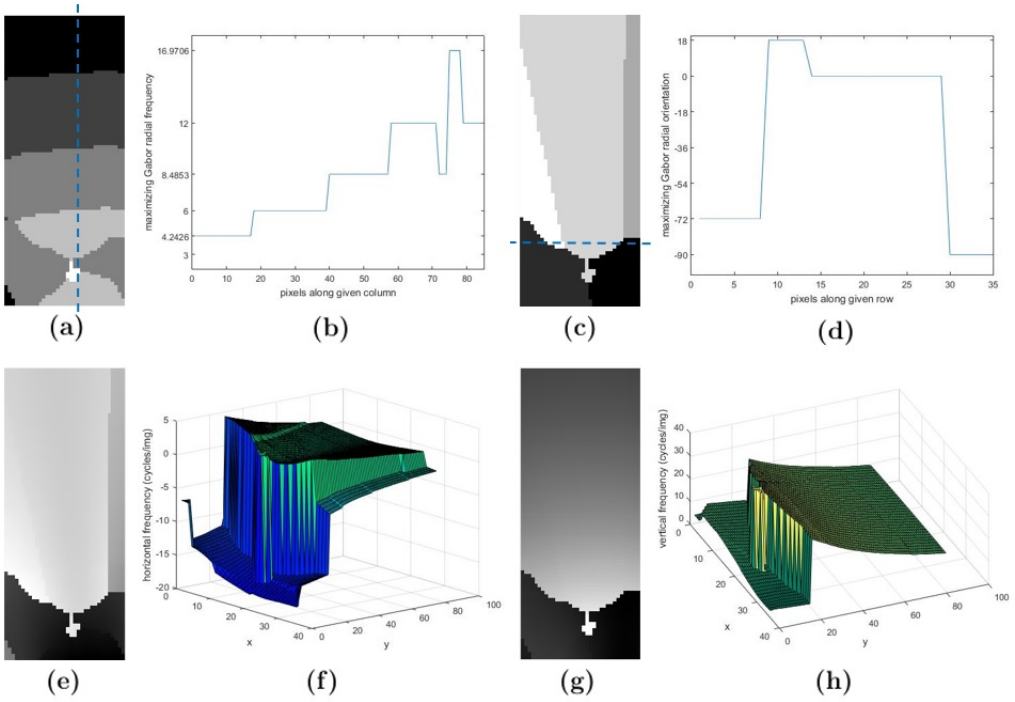


FIGURE 4.6: Closer look at drift in dominant instantaneous frequency estimate via demodulation. Radial center frequency (a) and orientation (c) of maximizing Gabor filter at each pixel. 1D plot (b) (respectively, (d)) of dotted line in (a) (respectively, (c)). Resulting dominant horizontal (e, f) and vertical (g, h) frequency estimates shown as 2D images and 3D surface plots.

other hand, the orientation plot reveals that the Gabors pre-dominantly fire strongly at the horizontal bars in the image (18°, 0°, −18° as one moves from left to right). However, in the lower region of the image, the vertical bars (−72°, 90°) are the ones that define the "dominant" Gabors. Fig. 4.6(e) and (g) show the resulting horizontal and vertical estimates obtained via Eqns. 4.15, followed by choosing the sign according to the quadrant of the maximizing Gabor. While the demodulation scheme recovers remarkably smooth estimates in the upper textured image region, which if free from outliers, the result in the lower region is affected due to drift. This results in severe discontinuities in the frequency estimates, as observed in the corresponding surface plots in Fig. 4.6 (f) and (h).

The manifestation of said discontinuity suggests that a possible resolution to the problem of drift may be obtained by enforcing smoothness via the following graph cut problem [8]:

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{\{p,q\} \in \mathcal{N}} V_{p,q}(f_p, f_q) \tag{4.18}$$

where $\mathcal{P}$ is the set of sites p to be labeled (pixels), and $\mathcal{N}$ is the set of all possible pairs of pixels (the 8-$\mathcal{N}$ system is employed in all experiments for this thesis). The set of labels $\mathcal{L}$ consists of the entire Gabor filter bank. The unary term D_p is defined as:

$$D_p(f_p) = \frac{\alpha}{A(f_p; p)} \tag{4.19}$$

where $A(\mathbf{u}; \mathbf{x})$ is as dictated by Eqn. 4.14, with $f_p = (\Omega_p, \theta_p) \in \mathcal{L}$ giving the filter with center frequency $\mathbf{u} = (\Omega_p \sin\theta_p,\ \Omega_p \cos\theta_p)$ at $\mathbf{x} = p$.

There are two ways in which the pairwise smoothness term $V_{p,q}$ may be defined. One approach is to force the labels Ω_p and θ_p to be smooth:

$$\begin{aligned} V_{p,q}(f_p, f_q) = V(f_p, f_q) = \beta(\Omega_p - \Omega_q).^2 \\ + \gamma(\sin\theta_p - \sin\theta_q)^2 \\ + \gamma(\cos\theta_p - \cos\theta_q)^2 \end{aligned} \tag{4.20}$$

In this scenario, demodulation (Eqns. 4.15) is performed *after* solving the problem 4.18 to obtain the optimal labeling f. Let us call this first approach Graph Cut Optimization (GCO). The affine rectification obtained using the resulting *optimal* frequency estimate is shown in Fig. 4.5(c). A substantial improvement over the non-optimal case (Fig. 4.5(b)) is seen. Fig. 4.7

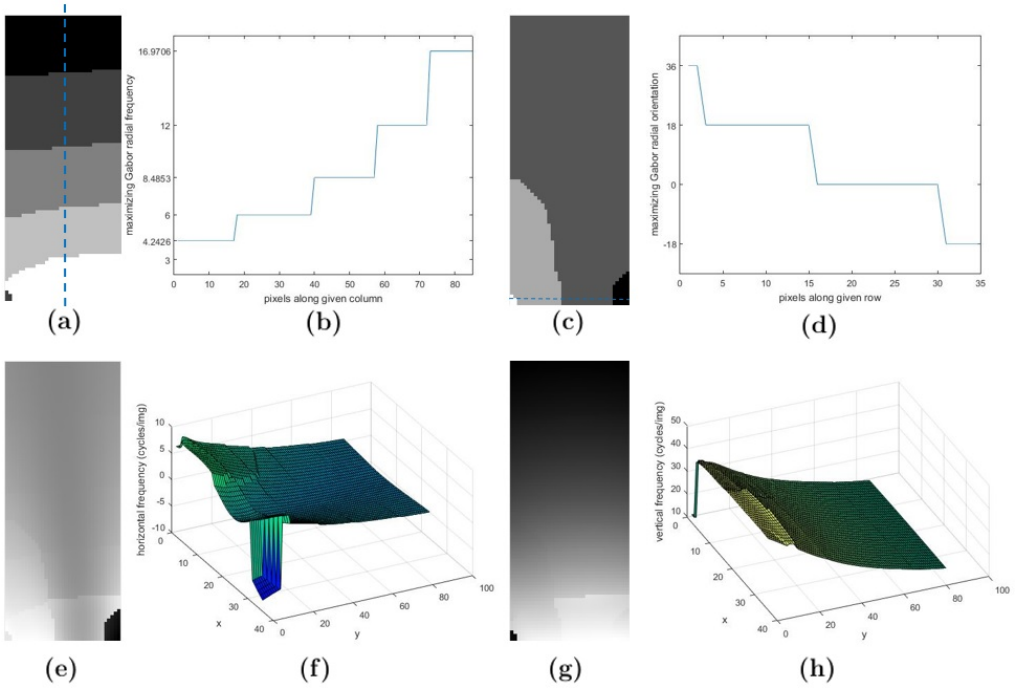


FIGURE 4.7: Resolution of frequency drift by enforcing smoothness over maximizing Gabor radial frequency, as well as over cosines and sines of radial orientation, via GCO. (a) – (h) same as Fig. 4.6

examines the optimal Gabor frequency and orientations obtained, as well as the resulting horizontal and vertical frequency estimates. A smooth, monotonically increasing frequency profile is observed in Fig. 4.7(b) along the sample dotted line in Fig. 4.7(a). Similarly, the orientations transit smoothly from 36° through to −18° along the dotted line in Fig. 4.7(c), as observed from Fig. 4.7(d). This indicates that Eqn. 4.18 helps to consistently track the varying frequency of the horizontally oriented bars in Fig. 4.5(a), and is not swayed by the vertical bars, even in the lower image regions.

Smoothing the sines and cosines in Eqn. 4.20 instead of the labels θ_p implicitly helps to recover smoother estimates of the horizontal and vertical frequency (Fig. 4.7(e – h)). Also, separating the radial frequency Ω_p and orientation θ_p terms in Eqn. 4.20 allows to fine-tune parameters β and

γ separately. In experiments, these parameters are fixed to $\beta = 1$ and $\gamma = 100$. For GCO, $\alpha = 1$ in Eqn. 4.19.

The second approach is to *explicitly* enforce smoothness on *signed* estimates $\tilde{u}(\mathbf{x})$ and $\tilde{v}(\mathbf{x})$ obtained *after* having applied demodulation (Eqns. 4.15) (with the sign defined by the candidate label Gabor's quadrant). That is, demodulation is performed for *all* labels (essentially, all Gabor filters) *first*, and an optimal Gabor is then obtained at each point such that the resulting labeling not only maximizes the response, but also yields smooth horizontal and vertical signed frequency estimates. The smoothness cost in this scenario, defined as:

$$\begin{aligned} V_{p,q}(f_p, f_q) = &\{\tilde{u}_{fp}(p) - \tilde{u}_{fq}(q)\}^2 \\ &+ \{\tilde{v}_{fp}(p) - \tilde{v}_{fq}(q)\}^2 \end{aligned} \tag{4.21}$$

is dependent on both the labels f_p as well as the sites p by virtue of the demodulation operation (Eqns. 4.15), which is site-dependent. Further, since the frequency estimates $\tilde{u}_{fp}(p)$ and $\tilde{v}_{fp}(p)$ can be arbitrary, the resulting energy 4.18 is non-submodular. We therefore employ quadratic pseudo-boolean optimization (QPBO) [66]. The α-expansion framework [8] is still used to handle the multiple labels, with QPBO as the sub-solver. QPBO can leave some nodes un-labeled in a given α-expansion iteration, and in such situations one may simply choose to retain the original labels of the affected nodes. The process is stopped if at any iteration there has been no reduction in energy. In practice, convergence is observed in around 2 – 6 iterations. For the 130x80 pixel example in Fig. 4.5(a), convergence was obtained in 3 iterations, with α set to 10^{-4} in Eqn. 4.19. The resulting

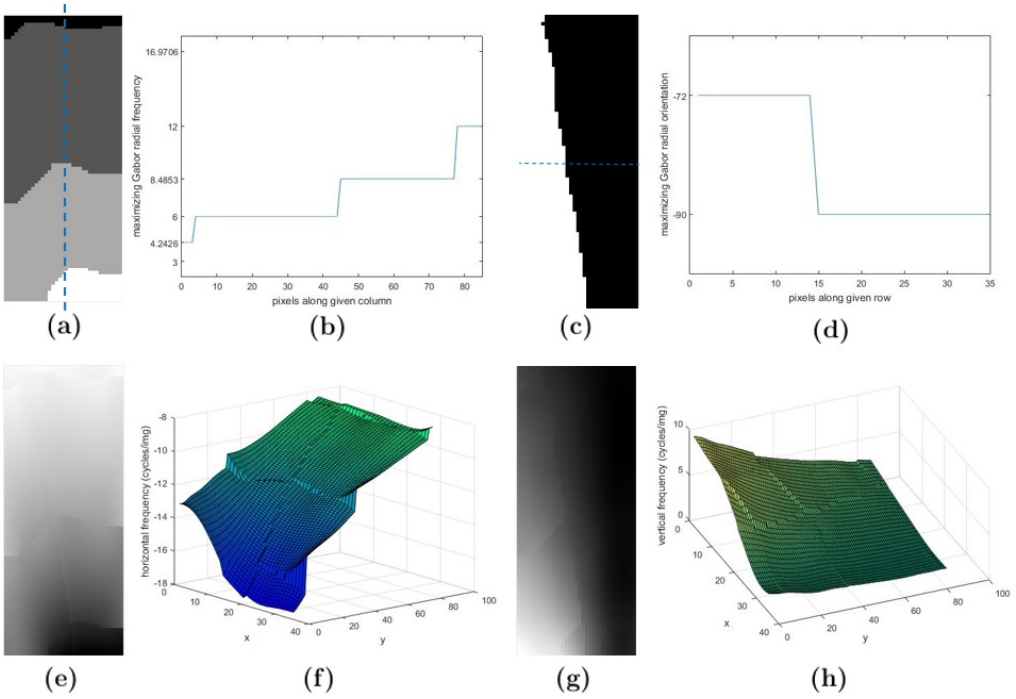


FIGURE 4.8: Resolution of frequency drift by enforcing smoothness via QPBO over dominant horizontal and vertical frequency components yielded by demodulation. (a) – (h) same as Fig. 4.6

	DEMOD	**GCO**	**QPBO**	**GT**
h7	0.2940	-0.0736	-0.0694	0.0089
h8	-0.2650	-0.4923	-0.4565	-0.6035

TABLE 4.1: Estimated projective parameters for the example texture in Fig. 4.5(a) using non-optimal frequency estimation (DEMOD), and the optimization based schemes (GCO and QPBO).

affine rectification is shown in Fig. 4.5(d) and is very similar to GCO [3] (Fig. 4.5(c)).

Fig. 4.8 illustrates the optimal radial frequency (a) and orientation (c), as well as the corresponding horizontal (e) and vertical (g) estimates obtained

[3]Both approaches to enforcing smoothness, i.e., 4.20 and 4.21, are essentially graph-cut problems — the first solved via max-flow, min-cut, and the second via QPBO. To differentiate between the two during discussion and for brevity, we use the term GCO to refer to the former, and QPBO for the latter. Also, these acronyms are more to refer to the method of smoothness in our context than the actual optimization algorithms.

by QPBO. Interestingly, for the example texture in Fig. 4.5(a), enforcing smoothness over the horizontal and vertical frequency *after* demodulation has the effect that the frequency of the vertically oriented bars is tracked, unlike DEMOD or GCO. This may be observed from Fig. 4.8(c – d), that show that Gabors oriented at $-72°$ and $-90°$ are chosen as optimal. Moreover, since it is the horizontal and vertical frequency obtained after demodulation that are smoothened, the estimates (Fig. 4.8(e – h)) are smoother than those obtained by GCO (Fig. 4.7(e – h)).

Nevertheless, the qualitative results in Fig. 4.5, and the recovered projective parameters (Table 4.1) indicate that both GCO and QPBO perform equally well. On the other hand, while GCO is fast (0.37s for this example), QPBO is considerably slower (2.53s). The main computational bottleneck is computing the smoothness term. The cost for GCO 4.20 may be computed once for each pair of labels. On the other hand, the cost for QPBO 4.21 depends on the labeling as well as the pair of pixels under consideration. Computing it all at once for every pair of neighbouring pixels and every possible label requires excessive memory. It must therefore be computed for every pair of pixels in every iteration of the alpha expansion loop using the current labeling.

4.4.2 Quadrant Ambiguity

Now consider the 80x160 pixel patch in Fig. 4.9(a) that is cropped from an image in the MIT Indoor67 category `subway`. In this example, the texture consists of the track rails that appear to converge as they recede from the camera. Fig. 4.9(b) shows the ground truth affine rectification. The DEMOD scheme in its original form again fails to work (Fig. 4.9(c)). However, if one rotates the given image counter clock-wise by 90°, uses

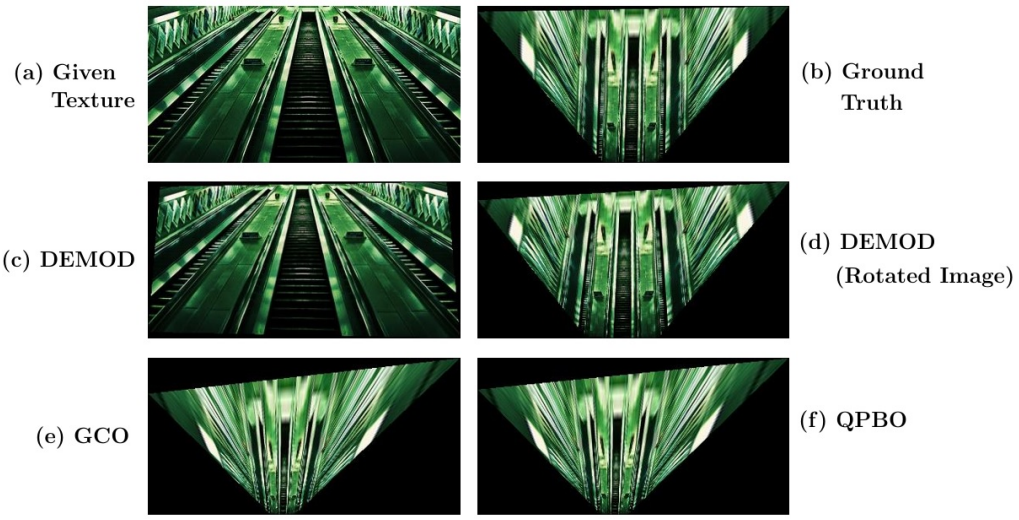


FIGURE 4.9: Affine rectification of given texture (a) via the model developed in Sec. 4.3 applied to dominant instantaneous frequency estimate. (b) Ground truth. Non-optimal estimate via demodulation (c) is prone to quadrant ambiguity, if manifested in given texture. (d) Demodulation applied to rotated texture does not face said ambiguity. Optimal estimates via GCO (e) or QPBO (f) can resolve any ambiguity.

DEMOD, and swaps the projective parameters so obtained (to cancel the effect of rotation), the resulting affine rectification is shown in Fig. 4.9(d).

The failure of DEMOD applied to the non-rotated patch may be understood, again, by inspecting the orientations of the dominant Gabor filters (Fig. 4.10, 2nd and 3rd rows). The orientation of the rails increases as one moves from left to right (36°, 54°, 72°), wraps around back to −90° (since we only sample two quadrants), and then increases again (−72° through to −36°). This is indeed the expected behaviour, but the resulting horizontal and vertical frequency estimates (Fig. 4.10, last two rows) suggest it is incorrect! The reason is that since we only sample frequencies from quadrants IV and I, a change from 72° to −90° results in a sharp discontinuity (a drop from +30 to -30 cycles/image) in the horizontal component (whose sign is dictated by the *sine* of the orientation — see Eqn. 4.17).

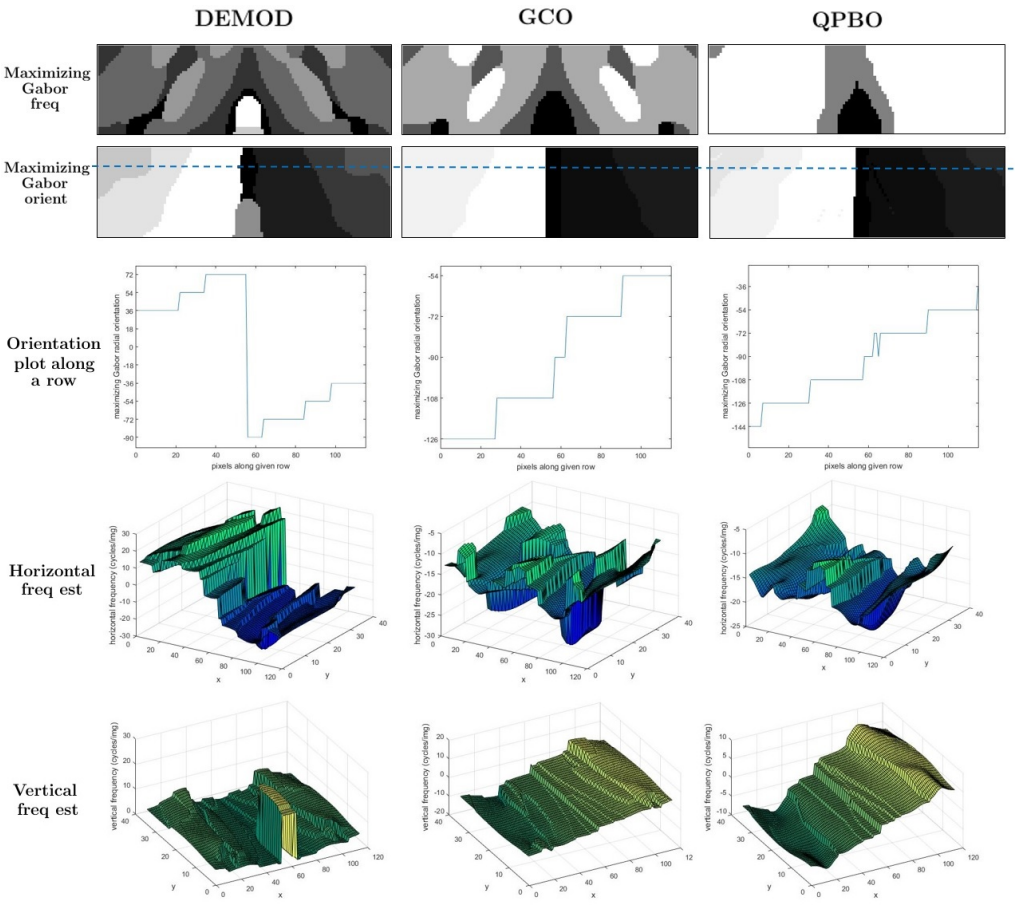


FIGURE 4.10: Closer look at quadrant ambiguity in dominant instantaneous frequency estimate via demodulation. Demodulation can only make use of quadrants IV and I; consequently, a change from $+72°$ to $-90°$ introduces a significant discontinuity in the horizontal frequency estimate. GCO and QPBO make use of all quadrants, ensuring a smooth transition from one quadrant to another with respect to both the horizontal and vertical frequency estimates.

In other words, any texture where the dominant frequency passes over from quadrant I to IV cannot be handled by DEMOD, unless the image is rotated! In practice, such texture abundantly appears on ceilings or floors in indoor scenes. We could swap our definition in Eqn. 4.17 such that the sign of the horizontal component is dictated by the *cos* function instead. However, any texture where the orientation passes over from quadrant IV to I will then face the same problem — such texture can appear on walls

	DEMOD	**DEMOD+ROT**	**GCO**	**QPBO**	**GT**
h7	-0.0360	0.0292	-0.0207	-0.0198	-0.0064
h8	-0.0422	0.6332	0.8509	0.8368	0.7011

TABLE 4.2: Estimated projective parameters for the example texture in Fig. 4.9(a) using non-optimal frequency estimation without (DEMOD) and with (DEMOD+ROT) rotation, and the optimization based schemes (GCO and QPBO).

in indoor scenes (see, e.g., Fig. 4.15(h)). A more principled approach is therefore needed to resolve the problem.

We again resort to enforcing smoothness via GCO and QPBO, as in the previous sub-section, except we now extend our set of labels $\mathcal{L}$ to consist of filters sampled at orientations from *all* the four quadrants. Note this is not possible with the original DEMOD scheme, since the corresponding frequency estimates from opposite quadrants have the *same* magnitude; hence, DEMOD cannot differentiate between a filter oriented at, say, 72° (quadrant I) and its counterpart at -108° (quadrant III) — there is an inherent *ambiguity* in assigning a quadrant to this filter. However, the demodulated frequency estimates resulting from these two filters do differ in signs, which may be exploted by GCO and QPBO. As illustrated in Fig. 4.10 (2nd and 3rd columns), the optimal orientations yielded by GCO and QPBO are those sampled from quadrant III and not I, thereby ensuring a smoother transition into quadrant IV with respect to both the demodulated horizontal and vertical frequency estimates (last two rows). The qualitative rectification results are given in Fig. 4.9(e) and (f) for GCO and QPBO respectively. Table 4.2 summarizes the recovered projective parameters for each approach.

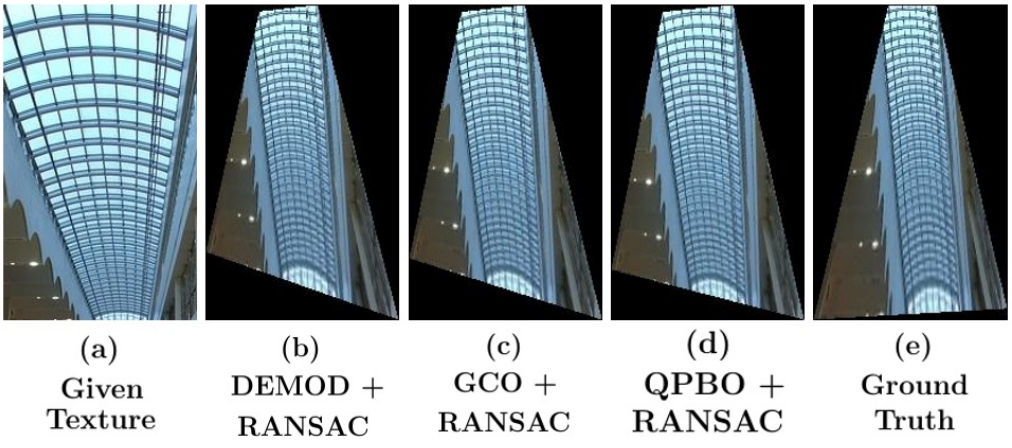


FIGURE 4.11: Improvement in affine rectification of texture with frequency drift via RANSAC based robust parameter estimation.

4.5 Robust Parameter Estimation via RANSAC

RANdom SAmple Consensus (RANSAC) [36] is a commonly employed approach to obtain robust estimates for model parameters when the measured data is noisy. Briefly, a random subset of data is picked consisting of the minimum number of points required to estimate the model parameters. A model instance is computed using this subset, and a 'consensus set' is then obtained from the data consisting of all inliers — points that are compatible with the estimated model within some pre-defined tolerance. A pre-defined number of iterations are performed to generate candidate sets of parameters. Then, the candidate that produces the largest consensus set is retained. A final estimate of parameters is then obtained using this entire consensus set. If, however, no iteration yields a sufficiently large consensus set, the algorithm reports a failure. RANSAC is a common tool in computer vision to robustly solve, e.g., for planar homographies in multi-view images in panoramic stitching, or for the fundamental matrix in stereo, etc.

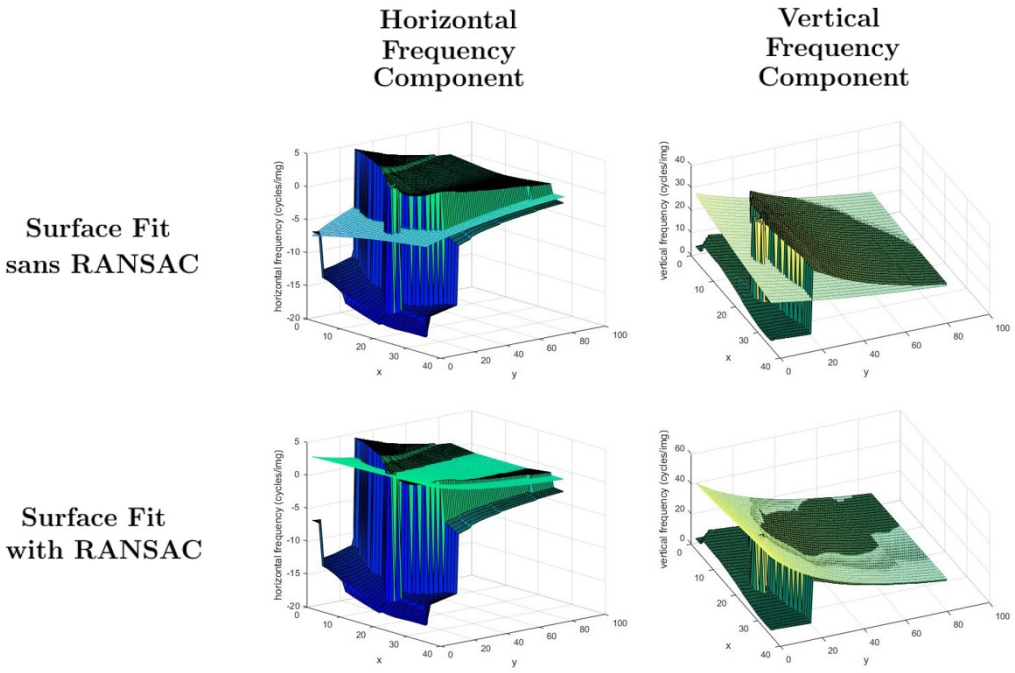


FIGURE 4.12: Robust parameter estimation via RANSAC rejects frequency drift as outliers.

In our setting of Eqns. 4.9, 4.11, a minimum of 2 points are sufficient to estimate the 4 parameters h_7, h_8, u_s, v_s. RANSAC is run for 50 iterations, with an error tolerance of 0.001, applied to Eqn. 4.12. The rectifications produced by the resulting robust estimates for the example image from Fig. 4.5 are presented in Fig. 4.11 for each frequency estimation scheme. Even the non-optimal DEMOD scheme produces a good affine rectification, that is comparable to GCO and QPBO. However, while DEMOD with RANSAC can seemingly handle frequency drift, as can be seen from Table 4.3, the percentage of outliers is significantly higher compared to GCO and QPBO. In Chapter 5, when we employ percentage of outliers as a metric to 'detect' homogeneous texture, that is where the optimization based approaches plus RANSAC yield better detection rates than DEMOD plus RANSAC, in the face of real world clutter.

	DEMOD	**GCO**	**QPBO**	**GT**
h7	-0.0750	-0.0733	-0.0646	0.0089
h8	-0.5267	-0.4962	-0.4577	-0.6035
hline % outliers	28.91%	8.07%	0%	N/A

TABLE 4.3: Robust estimated projective parameters for the example texture in Fig. 4.11(a) using non-optimal frequency estimation (DEMOD), and the optimization based schemes (GCO and QPBO). The percentage of RANSAC outliers is also reported. RANSAC error tolerance = 0.001.

Fig. 4.12 provides a visual illustration of how RANSAC can improve parameter estimation. The estimated constant frequency u_s, v_s is re-projected using the estimated parameters h_7, h_8, and the resulting mesh is drawn on the same plot as the surface showing the demodulated frequency. Without RANSAC, the parameters are bogged down by outliers, whereas a robust estimation of parameters can reject outliers.

Fig. 4.13 shows affine rectifications via robust parameter estimation for the example with quadrant ambiguity using the various frequency estimation schemes. We observe that since the proportion of outliers in this example is large — the inliers and outliers are roughly divided 50/50 (see, Fig. 4.10, left-most column) — RANSAC is only able to produce a partial rectification. Table 4.4 reports the estimated parameters along with percentage of outliers in each case. A RANSAC error tolerance of 0.01 was used; a stricter threshold of 0.001 resulted in failure (i.e. $> 50\%$ outliers in each case).

4.6 Anisotropic Multiscale Representation

It must be noted that this frequency based rectification pipeline is highly sensitive to parameters such as filter size and image patch size. Extensive

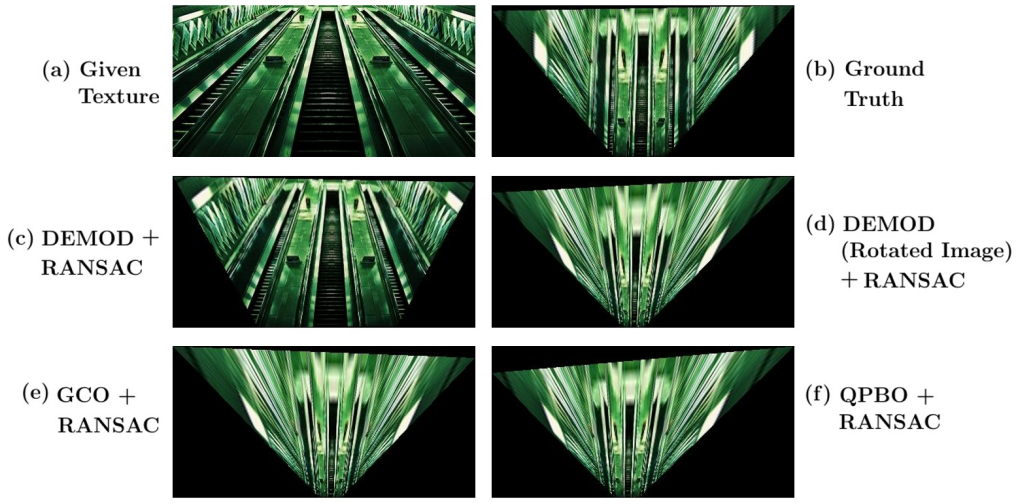


FIGURE 4.13: RANSAC struggles to overcome quadrant ambiguity (c) if proportion of outliers is large.

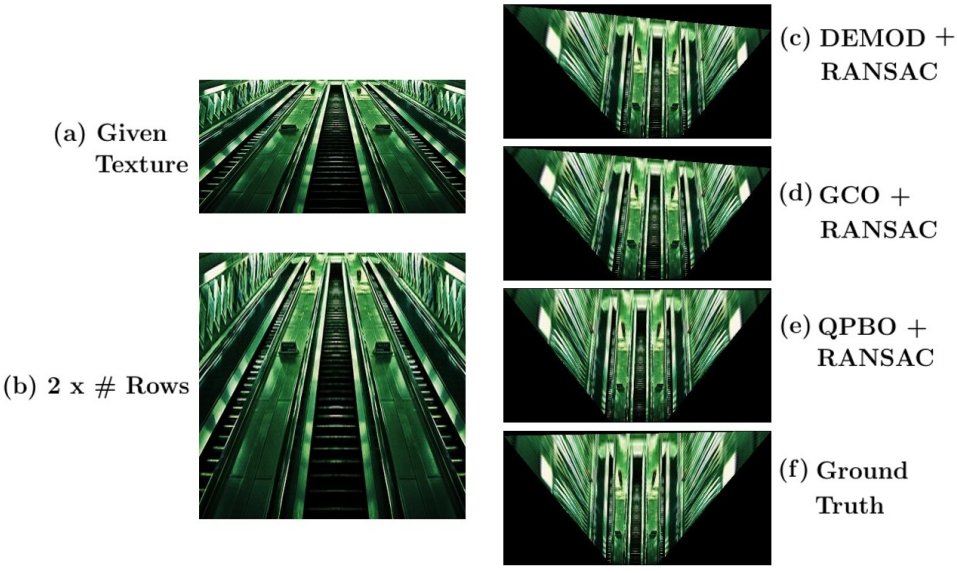


FIGURE 4.14: An anisotropic multi-scale approach, combined with carefully normalized error measure for choosing the best scale, improves texture rectification. Rotation may be allowed for DEMOD to automatically resolve quadrant ambiguity, if any.

experiments in the course of this thesis have helped to fine-tune parameters that yield the overall best results. The filter bank parameters have been

	DEMOD	**DEMOD+ROT**	**GCO**	**QPBO**	**GT**
h7	0.0190	-0.0081	0.0059	-0.0160	-0.0064
h8	0.3674	0.8695	0.8616	0.8475	0.7011
% outliers	48.42%	3.48%	3.45%	0.45%	N/A

TABLE 4.4: Robust estimated projective parameters for the example texture in Fig. 4.13(a) using non-optimal frequency estimation without (DEMOD) and with (DEMOD+ROT) rotation, and the optimization based schemes (GCO and QPBO). RANSAC error tolerance = 0.01.

described in detail in Sec. 4.4, with the filter kernel size fixed to 45x45 pixels. Meanwhile, the image patch to be filtered should be resized such that the smaller dimension is 80 pixels (using bicubic interpolation), and the aspect ratio is retained. The partial derivatives needed for demodulation (Eqns. 4.15) were obtained via a simple forward difference approximation on the texture image. A Central difference approximation, or the use of filter masks involving it — e.g., Sobel and Fri-Chen — can only successfully recover half of the otherwise maximum measurable frequency, due to aliasing. This was observed in texture containing high frequency, where measuring changes over each pixel counts (see, e.g., Fig. 4.15(g)). Following [123], the filter responses are smoothened by a Gaussian low-pass filter, also sized 45x45 pixels, and having a standard deviation $1/12^{th}$ its size.

It was additionally observed that a anisotropic multi-scale approach improves rectification. The given image is represented at three scales — one where the smaller dimension is 80 pixels, second where the rows are doubled while columns stay the same, and third where columns are doubled and rows stay the same (bicubic interpolation is used for the required resizing). For e.g., the `subway` patch is originally 200x400 pixels. It is resized to give three representations: 80x160 pixels (shown in Fig. 4.14(a)), 160x160 pixels (shown in Fig. 4.14(b)) and 80x320 pixels. Parameters are obtained for each representation, and the one that results in the largest percentage of

RANSAC inliers defines the winning parameters. The resulting affine rectifications are shown in Fig. 4.14(c — e) for DEMOD, GCO and QPBO, respectively. In each case, the winning representation was determined automatically, and happened to be case# 2 — i.e., doubling of rows (Fig. 4.14(b)). Moreover, for DEMOD, rotated patches were also included to handle quadrant ambiguity, giving six representations in total (the winning representation happened to be a rotated version with double the rows, i.e., Fig. 4.14(b)). The anisotropic scaling essentially makes the scale of the relevant image features (track rails in our example) more pertinent with respect to the size of the Gabor filters used (45x45 pixels).

Finally, it should be noted that our error measures in Eqns. 4.9 and 4.11 are not defined in the euclidean space, but in a non-linear and an affine-transformed space, respectively. As such, it is not meaningful to compare them across patches or across scaled representations of a given patch, in either deciding what threshold to set beyond which a patch is deemed non-homogeneous for the former, or in choosing a winner among different scaled representations of a patch for the latter. Formally, the error is not affine invariant.[4] In this regard, the following heuristic normalization approach was observed to produce the best results. RANSAC is first performed using a fixed error threshold of 0.001 on Eqn. 4.12 to obtain a robust estimate of parameters as well as the best set of inliers. The dynamic range of the radial frequency estimate is computed using these inliers as:

$$\mathcal{DR} = \max_{i \in inliers} (\tilde{\mathbf{u}}_\mathbf{i}) - \min_{i \in inliers} (\tilde{\mathbf{u}}_\mathbf{i}) \tag{4.22}$$

[4]The Fourier spectrum (magnitude of the Fourier transform) of a given texture is known to be invariant to an affine transform upon normalization by its l_1-norm [147]. Our scenario, however, concerns the frequency plane *coordinates* (i.e., the frequency itself), having undergone said unknown transform.

	DEMOD+ROT	**GCO**	**QPBO**	**GT**
h7	0.0437	0.0431	0.0025	-0.0064
h8	0.6563	0.6087	0.6597	0.7011
% outliers	46.24%	32.52%	16.18%	N/A

TABLE 4.5: Robust estimated projective parameters for the example texture in Fig. 4.14(a) using an anisotropic multi-scale approach for DEMOD+ROT, GCO and QPBO. RANSAC error tolerance = 0.001.

where, $\tilde{\mathbf{u}}_\mathbf{i} = \tilde{\mathbf{u}}(\mathbf{x_i}) = \sqrt{\tilde{u_i}^2 + \tilde{v_i}^2}$ is the radial frequency estimate. A normalized residual re-projection error is then computed for all points $\mathbf{x_i}$, i.e. inliers as well as outliers:

$$\mathcal{E}(\mathbf{x_i}) = \frac{\tilde{\mathbf{u}}(\mathbf{x_i}) - \mathbf{J'_P}(\mathbf{x_i})\mathbf{u_s}}{DR} \tag{4.23}$$

where $\mathbf{J'_P}(\mathbf{x_i})$ re-projects the robust estimate of intermediate plane frequency $\mathbf{u_s}$ to the image plane (see Eqn. 4.8). The *normalized* root mean squared error (RMSE) is then:

$$\mathcal{RMSE} = \sqrt{\sum_i \mathcal{E}(\mathbf{x_i})} \tag{4.24}$$

For a multi-scale representation giving $> 50\%$ outliers, the normalized error is set to infinity. In Fig. 4.14, the winning multiscale representation for each frequency estimation scheme was obtained based on the normalized RMSE 4.24. The qualitative as well as the quantitative results (summarized in Table 4.5) indicate a marked improvement over a uni-scale approach.

4.7 Results and Comparisons

This section evaluates the affine rectification scheme proposed in this chapter, and compares it with two representative methods in literature —

Transform-Invariant Low-rank Texture (TILT) [149], and Repetition Maximization (REM) [3]. The evaluation is based on $N = 30$ patches, cropped from various images in MIT Indoor67, depicting some homogeneous texture under perspective projection. Qualitative results (Sec. 4.7.1) are included for about half the test cases to save space, while the quantitative results (Sec. 4.7.2) take all 30 test cases into account. A brief description for each scheme appears below.

TILT [149]: The code made available online by the authors is employed with default settings. It implements a multi-scale approach, and automatically localizes a region of interest that it senses to depict a low-rank texture in order to recover the projective parameters.

REM [3]: A demo command-line program made available online by the authors is used — allowing a multi-scale search — to generate the qualitative results. The estimated parameters are not returned, however, so a quantitative comparison with REM is not performed.

DEMOD: The dominant frequency estimation method in its original form is employed, as given in [123] and reviewed in Sec. 4.4, while the texture projection model developed in Sec. 4.3 is used to obtain projective parameters. RANSAC is not applied.

RANSAC: Same as DEMOD with RANSAC (Sec. 4.5) applied. Additionally, an anisotropic multi-scale approach is used, and rotation is allowed (Sec. 4.6).

GCO: Graph-cut optimization with smoothness enforced on filter radial frequencies as well as the *sine* and *cosine* of filter radial orientations, solved by alpha-expansion, followed by demodulation (Sec. 4.4). RANSAC is used for robust parameter estimation, and an anisotropic multi-scale representation is used.

QPBO: Graph-cut optimization with smoothness enforced on horizontal and vertical frequency estimates obtained upon demodulation, optimized via QPBO (Sec. 4.4). RANSAC is used for robust parameter estimation, and an anisotropic multi-scale representation is used.

4.7.1 Qualitative Performance

Figs. 4.15 present the results for affine rectification. The various examples also help to appreciate the ubiquitous presence of homogeneous texture in indoor scenes.

It can be observed that TILT in general performs well only in a limited number of cases, where the underlying texture is low-rank, with few outliers, e.g., (a) and (b). In situations where the texture departs from the low-rank assumption — e.g., port-holes (d), or barrels (e), where the gradients are isotropic in all directions — TILT cannot be expected to perform. For case (e), TILT returns a vanishing line (essentially, $[h_7\ h_8\ 1]$) that passes through the image patch, and thus a distorted rectification results. On the other hand, the frequency based schemes are seen to handle such texture very well, corroborating our intuition that homogeneity is a more general assumption than low-rankness.

TILT also breaks when the noise is not sparse, and this is very common in real-world indoor scenes. For e.g., the `airport_inside` ceiling (c), where the texture has a limited spatial support. Or the brick wall in (g) where it likely fails due to outliers with large spatial support, significantly corrupting the texture. Another failure case for TILT is the ceiling in (g), which, albeit low-rank, also manifests significant outliers.

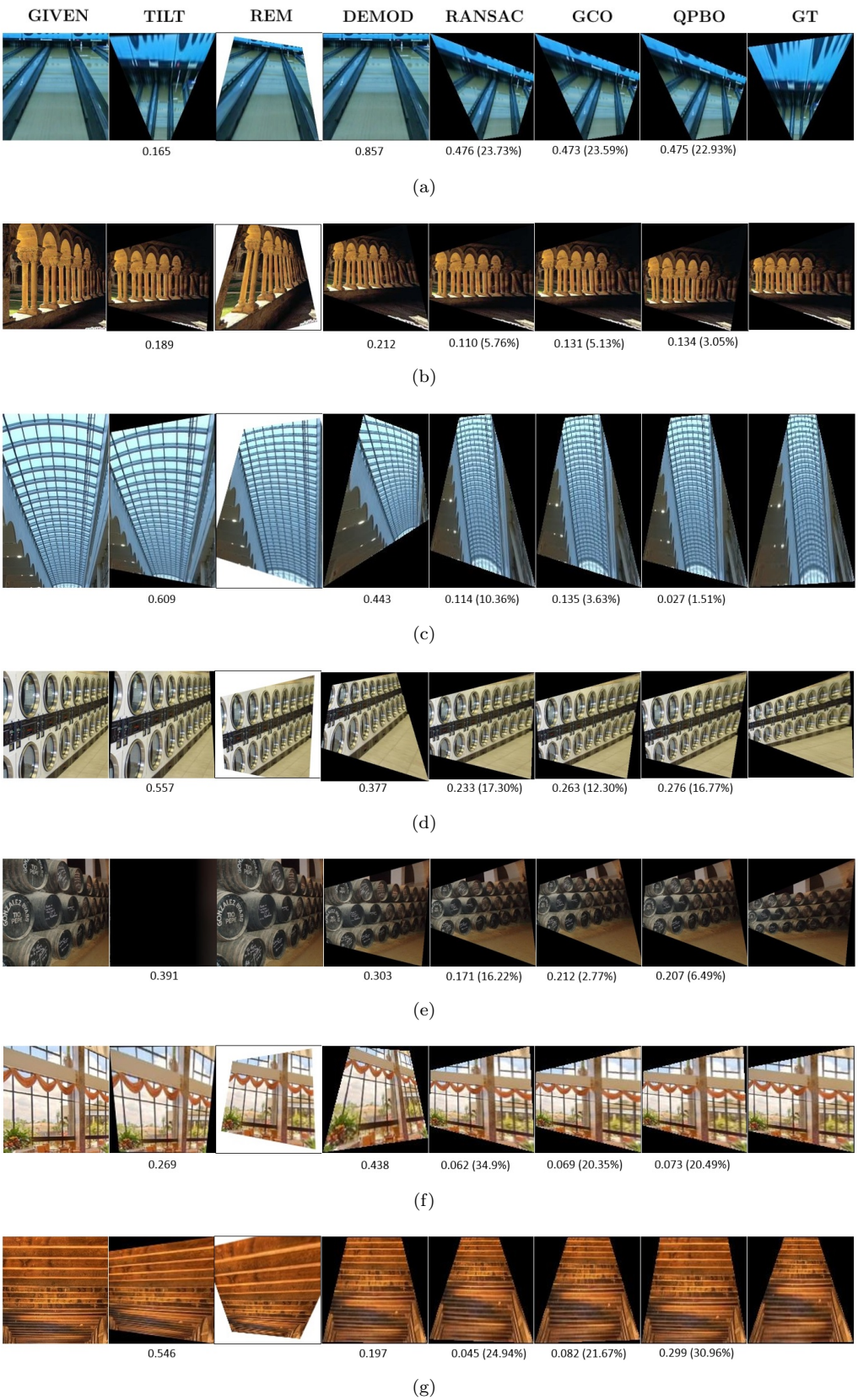
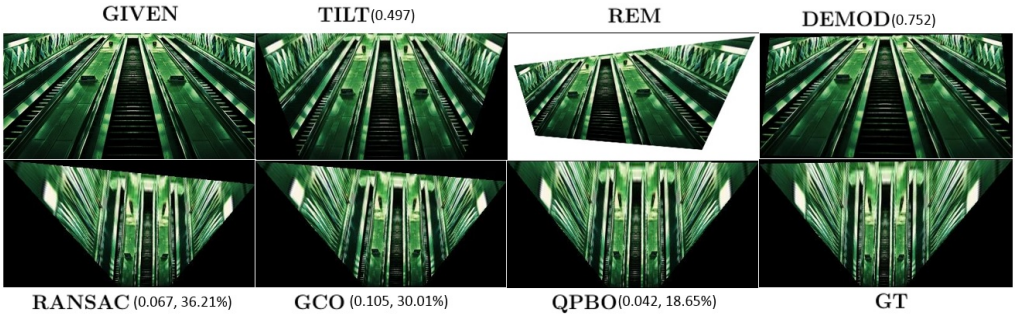
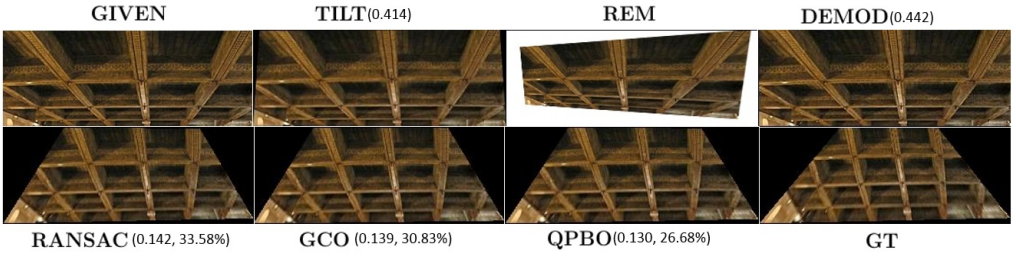


FIGURE 4.15: Qualitative results for affine texture rectification — 1/3. Author implementations for TILT [149] and Repetition Maximization (REM) [3] have been used. Estimation error (Eqn. 4.25) is also reported (except REM), and %outliers for RANSAC, GCO and QPBO.

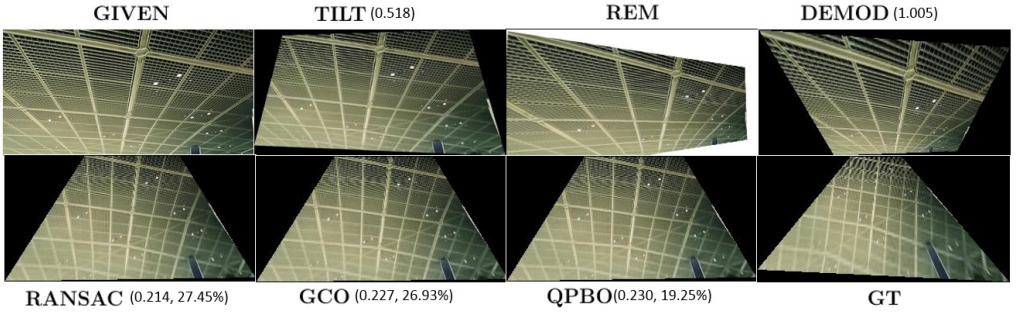
FIGURE 4.15: Qualitative results for affine texture rectification — 2/3. Author implementations for TILT [149] and Repetition Maximization (REM) [3] have been used. Estimation error (Eqn. 4.25) is also reported (except REM), and %outliers for RANSAC, GCO and QPBO.



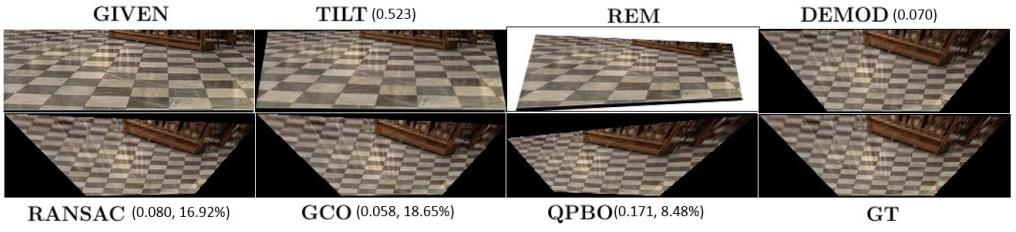
(n)



(o)



(p)



(q)

FIGURE 4.15: Qualitative results for affine texture rectification — 3/3. Author implementations for TILT [149] and Repetition Maximization (REM) [3] have been used. Estimation error (Eqn. 4.25) is also reported (except REM), and %outliers for RANSAC, GCO and QPBO.

METHOD / METRIC	TILT	DEMOD	RANSAC	GCO	QPBO
MEAN EST. ERROR	0.496	0.386	0.190	0.186	0.187
% OF OUTLIERS	N/A	N/A	25%	20.76%	18.39%

TABLE 4.6: Affine rectification — quantitative evaluation. RANSAC error tolerance = 0.001.

Both TILT and REM can be seen to perform poorly in cases with illumination changes (l, n, q). On the other hand, use of Gabor filters allows the frequency based schemes to perform remarkably well in these challenging cases. Provided the scale of texture is small (i.e., texture contains higher frequencies) relative to the scale of the surface it covers, a frequency based representation is resilient to slow-varying (low-frequency) photometric changes (see [123]). TILT and REM also seem to fail on cases exhibiting large perspective distortion, e.g., the textured ceilings in cases (o, p). The ground truth for (p) shows the patch may not be uni-planar, hence it is not strictly low-rank or even homogeneous. Nevertheless, the robust frequency based schemes perform favorably.

REM — which has only been demonstrated for properly cropped, printed patterns — seems to rarely perform well on our challenging cases that exhibit limited spatial support, significant clutter and illumination changes.

4.7.2 Quantitative Performance

For a quantitative evaluation, the following metric is used:

$$\textbf{Mean Estimation Error} = \sum_{i=1}^{N} \sqrt{(\tilde{h_{7i}} - h_{7i})^2 + (\tilde{h_{8i}} - h_{8i})^2} \qquad (4.25)$$

where $\tilde{h_{7i}}$, $\tilde{h_{8i}}$ are the parameters returned by an algorithm, and h_7, h_8 are the ground truth parameters obtained by manual annotation of vanishing points. $N = 30$ is the number of test cases used.

The results are summarized in Table 4.6. Interestingly, TILT performs worse than DEMOD — which is what this chapter has proposed improvements for. GCO and QPBO perform equally in terms of Mean Estimation Error, with RANSAC performing slightly worse off. GCO and QPBO resolve frequency drift and any quadrant ambiguity by imposing smoothness priors and proposing robust frequency estimates. On the other hand, RANSAC overcomes drift by rejecting outliers, and employs both the original and rotated images to decide the best parameters, thereby resolving quadrant ambiguity, if any.

However, as was also observed previously in Sec. 4.5, the percentage of outliers can serve as a suitable metric to detect homogenous texture under perspective projection. While for a *known* homogeneous texture one may altogether forego robust optimization based frequency estimation, GCO or QPBO are indispensable for a detection pipeline. Also note that while the percentage of outliers can in principle be computed from TILT by looking at the support of the sparse outlier matrix, we do not do so as it is tangential to our interest.

Chapter 5

Detection of Homogeneous Texture in Indoor Scenes & its Geometric Class Assignment

Sec. 5.1 motivates the detection of homogeneous texture in indoor scenes as useful mid-level features for recognition that are additionally invariant to viewpoint changes, and highlights the merits of such an approach over others in literature. Sec. 5.2 performs said detection on the MIT Indoor67 dataset, and qualitatively analyzes and compares the detections for some example images with an existing work (TILT [149]). Sec. 5.3 shows that it is possible to estimate a spatial layout in scenes with a sufficiently abundant presence of regular texture. A comprehensive evaluation is presented based on qualitative results, contrasting the pros and cons with an existing approach (see Sec. 3.2.1) that exploits scene vanishing points and machine learning. The discussion lends useful insights into the workings of the proposed approach. Sec. 5.4 suggests that if scene vanishing points are known,

it is possible to upgrade the affine rectification to metric rectification. Finally, Sec. 5.5 presents a quantitative evaluation of the proposed detection, demonstrating its superior performance over TILT.

5.1 Background

Since indoor scenes can be well described by the objects and components they contain, indoor scene recognition has typically been approached through the detection of class-discriminative, mid-level visual features or parts that preserve semantics and spatial information (Sec. 2.6). Automatic learning of such representative and discriminative parts from images, labeled only with the scene category, has received wide attention [95, 119, 63, 23]. As discussed in Sec. 3.1.2, however, the problem is ill-posed, since neither part instances nor part models are known beforehand.

The alternative approach is to employ hand-crafted detectors that do not require learning from weakly labelled and limited training data. Existing work on feature detection (Sec. 2.3.1), however, caters only to detecting prominent or salient local, *low-level* interest regions such as edges, curves or blobs. It was reviewed in Sec. 2.3 that sparse scene representations resulting from these low-level detections perform poorly compared to dense representations when it comes to recognition. This is because local interest region detection is prone to pre-maturely discarding discriminative scene information. On the other hand, the survey in Sec. 2.6 suggested that sparse representations based on *mid-level* features can perform very well, and in fact are complementary to local dense features. The reason is that mid-level features not only capture scene semantics, but also afford better intra-class invariance as opposed to low-level features.

Our goal in this chapter is therefore also to detect mid-level, semantically meaningful regions. However, instead of learning a host of individual part models for representative scene regions, we would like to exploit the ubiquity of a generic mid-level visual attribute in indoor scenes — homogeneous texture. Numerous examples of such texture have been presented in Chapter 4. Fig. 5.1 depicts some additional and very interesting cases that commonly manifest in indoor scenes (again, from the MIT Indoor67). Ceilings in indoor pools, greenhouses or courtyards very often exhibit uniform woodwork (a), or engraved and printed patterns (b). Ceiling lights in grand venues such as concert halls or theaters, or in large hallways occur in patterns (c). Repeating columns and pillars are characteristic of enclosed walkways, underground cellars or expansive indoor spaces such as a train station (d). Even uniformly laden tableware and buffet items (e), row-planting and well-arranged floristry satisfy homogeneity (f). The furniture itself can often exhibit uniform patterns — casino kiosks, a cluster of computers, and aligned chairs in dining rooms and classrooms are a few examples (g). Grillworks and railings (h) are a common indoor feature. Surprisingly, even shadows can give rise to homogeneous texture (i), provided the causing obstructions such as columns and walls are uniform in arrangement, and the surface is planar! Awe-inspiring interiors characteristic of airports and subways often happen to be patterned (j), and so are most window arrangements in any indoor environment (k).

While man-made indoor scenes are full of such regular patterns, they appear at unknown spatial locations, scales and viewpoints. In addition, real-world indoor scenes are fraught with unwanted interference such as noise, room clutter, and varying lighting or illumination effects over a given texture (see Figs. 4.1, 4.2). Furthermore, the wide variety of such homogeneous texture necessarily entails large variation across instances — the repeating "texels"

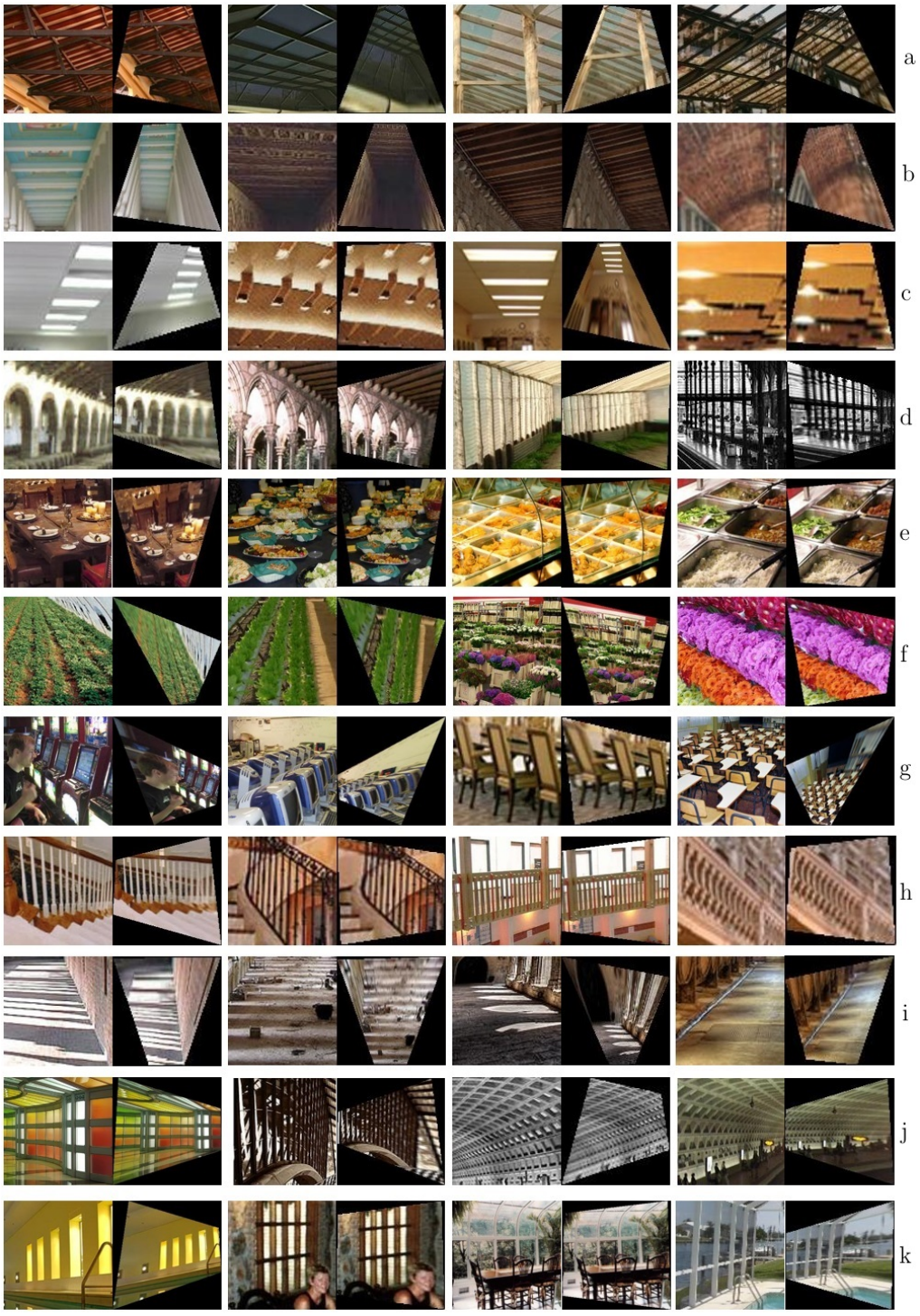


FIGURE 5.1: Abundantly present and variedly manifested, homogeneous texture in indoor scenes can serve as useful mid-level features for recognition; both architectural structure as well as scene contents exhibit homogeneity. All depicted texture was detected and rectified automatically via the proposed approach.

or "motifs" can virtually take on *any*, unknown form (from shadows to dinner plates)! The daunting task of localizing meaningful patterns in indoor scenes in the presence of such out-of-control factors is, therefore, that of "detection in the wild".

A previous attempt by TILT [149] has been made to localize low-rank texture in a given single-object image, or to detect instances in a highly textured urban scene with no clutter. However, the evaluation for affine rectification presented in Sec. 4.7 suggests that TILT is not sufficiently robust to take on the above challenges. The frequency based texture rectification model developed in Chapter 4, however, equipped with robust dominant frequency estimation (Sec. 4.4) and robust parameter estimation (Sec. 4.5), was observed to perform remarkably well in the face of outliers, clutter and photometric changes (see Sec. 4.7.1). The use of a generic Gabor filter bank (as opposed to using low-level feature detectors) lends itself well to describing any form of homogeneous texture. Sec. 5.2, therefore, puts this model to use for the aforementioned problem of detection. In doing so, no iterative learning of region-specific models is needed, and the approach is therefore not affected by the limited availability of training data. Sec. 5.2.3 and 5.5 present, respectively, qualitative and quantitative evaluations of the proposed approach, and TILT will again be seen to perform poorly in comparison.

Part based representations are not invariant to affine geometric (though invariance to uniform scale is incorporated via multi-scale detection), let alone the more general projective transforms. A part based representation for a given scene region, therefore, must learn separate models for different viewpoints anew, while invariance to appearance variation depends on how well a particular part *discovery* algorithm and the descriptor employed can generalize to similar regions (which is not trivial, given the ill-posed nature

of the problem). On the other hand, region detectors such as [88, 64] afford local affine invariance, but they only yield low-level edge and blob-like features. So while the concept of rotation invariance [84], and the more general affine adaptation [88, 64] exists in literature for low-level features, projective rectification has never been employed. This is probably because the need was never felt — for an image region with small dimensions compared to its depth from the camera, perspective effects may be approximated by an affine model. The texture projection model developed in Chapter 4, however, explicitly caters to projective transforms in meaningful *mid-level* image regions, and was consequently observed to overcome *significant* perspective distortions (see Figs. 4.1, 4.15). Note that although the resulting *rectification* is within an affinity of the world plane (i.e., affine geometric change is not recovered), the *detection* per se (Sec. 5.2) is invariant to projective transforms (which subsume affine transforms). Furthermore, the model is capable of detecting *any*, generic, homogeneous texture, and is therefore *highly* invariant to appearance changes within this rather rich class of meaningful scene regions. Chapter 6 makes use of the resulting detections and rectifications to push scene recognition performance.

5.2 Detection in the Wild

In Sec. 4.6, a method was devised to choose the best rectification parameters from among different scaled representations for a given, *known* texture patch. Briefly, RANSAC is used to fit robust model parameters to dominant instantaneous frequency estimates for each representation. The dynamic range of frequency of the resulting inliers is used to normalize the residual re-projection error (based on the recovered model parameters). The scaled

representation yielding the lowest normalized RMSE (Eqn. 4.24) defines the winning parameters.

We may adopt a similar criterion to *decide* whether a given image patch depicts homogeneous texture. Specifically, if the resulting RMSE is below a certain threshold, it is admitted as containing homogeneous texture. Equivalently, a percentage of outliers may be computed, such that points $\mathbf{x_i}$ having a normalized squared error $\mathcal{E}^2(\mathbf{x_i})$ (Eqn. 4.23) larger than a certain threshold (fixed at 0.01 for all experiments in this chapter) are deemed as outliers. Then, an image patch is accepted as depicting homogeneous texture if it contains fewer than a given percentage of outliers (set to 50% for all experiments in this section), thereby making for a more intuitive detection metric.

5.2.1 Scale-Invariant Detection

An approach similar to multiscale object detection [33, 119] is taken, wherein a given image is represented at multiple scales, and patches of fixed size extracted and processed at each scale. This provides for a space and scale invariant detection.

Specifically, a given image is first resized to a reference scale, such that the smaller dimension is 400 pixels, and the aspect ratio preserved. Patches, sized 80x80 pixels, are extracted on a regular grid with a spatial stride of 16 pixels. This gives the number of octaves, such that at least one such patch may be extracted at the coarsest scale, as $log_2(400/80) = 2.3$. Fixing the number of scales per octave to 3.5, the total number of levels in our multiscale pyramid is then $N = floor(2.3 \times 3.5) + 1 = 9$. The corresponding scales to resize the image to (via bicubic interpolation) are given by a geometric progression with common ratio $r = 2^{-1/3.5}$, i.e., r^l,

where $l = 0, 1, ..., N-1$. Following [119, 63], a patch containing very little image variation, i.e., gradient energy (average gradient norm over all pixels) smaller than a certain threshold (fixed to 50% of the average gradient energy over all image patches) are discarded at the outset. This results in a total of around 1500 patches per image on average. A smaller grid spacing may be used at higher computational expense (e.g., a spacing of 8 pixels can result in four times the number of patches). Also, a non-unit aspect ratio for patches (e.g., sizes of 80x160 or 160x80, etc) can often be more representative of the homogeneous texture occurring in scenes, and sampling such additional patches to improve detection and recognition performance may be done at higher computational expense.

5.2.2 Other Implementation Details

For the qualitative results presented in this chapter, an **intra-scale non-max suppression (NMS)** is performed as follows. Candidate patches (those with $< 50\%$ outliers) are sorted and processed in ascending order of percentage of outliers. Then, a patch is admitted as a detection only if some previously admitted patch (detected at the same scale) does not overlap 50% of its area. NMS across scales tends to discourage detections at coarse scales, hence suppression only within a given scale is carried out.

The Gabor filter bank as constructed in Sec. 4.4, consisting of 6 radial frequencies and 10 radial orientations, with the filter kernel sized 45x45 pixels is used. As discussed in Sec. 4.6, partial derivatives are computed via forward difference approximation, and the filter responses smoothened by a Gaussian. Rather than convolving Gabors with each patch individually, the entire image is convolved, followed by extracting filter responses at the corresponding patch locations. This considerably speeds up the process,

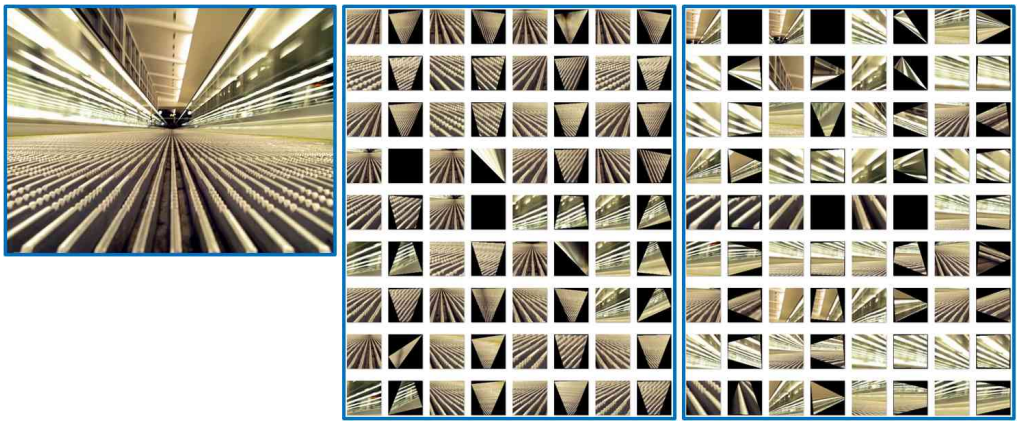
since redundant filtering is avoided (patches are overlapping in space; this does not lead to a difference in performance). The GCO (with RANSAC) configuration as described in Sec. 4.7 is used. As observed in Chapter 4, GCO performs similar to QPBO yet is considerably faster, while an approach based solely on RANSAC would report high proportion of outliers even for patches that do contain homogeneous texture.

While the experiments in Sec. 4.7 ran 50 iterations of RANSAC, the ones in this chapter use an adaptive scheme where the maximum number of iterations to run is updated continuously based on the current proportion of outliers in a given iteration [36]. RANSAC can then terminate in much fewer iterations. While this speeds up the process, using more RANSAC iterations would likely improve performance. Since we process a large number of overlapping patches, however, we may choose to make this trade-off.

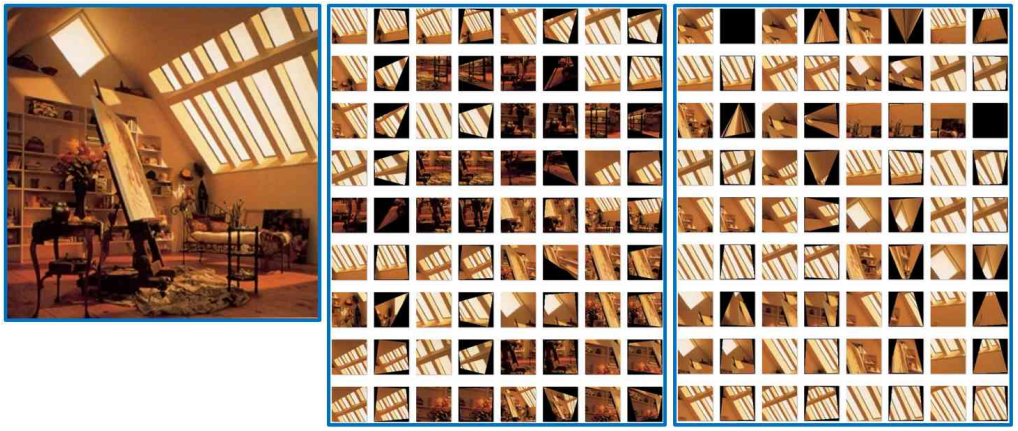
Given the experimental set-up as described above, processing one image takes around 15 – 20 mins per CPU core running a MATLAB implementation at 3GHz.

5.2.3 Discussion

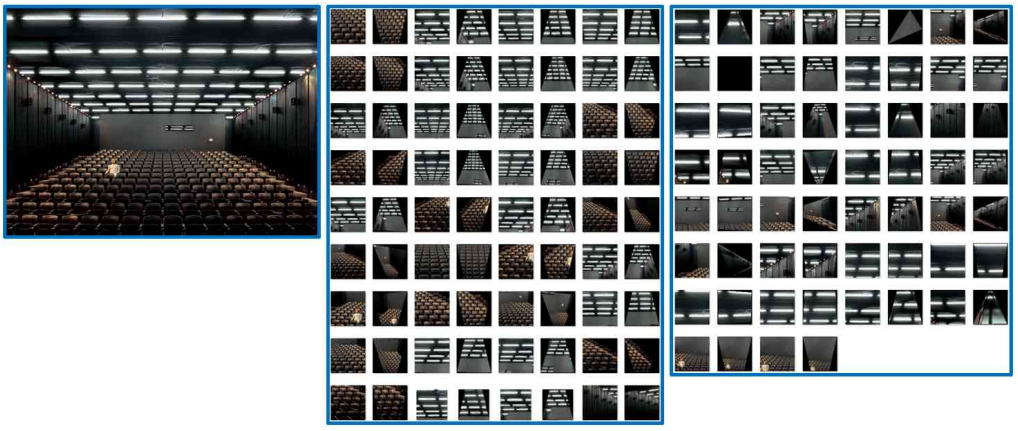
Fig. 5.2 presents a qualitative comparison of the proposed homogeneous texture detection vs. that performed by TILT [149] on a number of MIT Indoor67 scene categories. The decision score for TILT used is a rank ratio of 0.5 (i.e., ratio of final to initial rank), along with the intra-scale NMS described in Sec. 5.2.2. Top detections are shown for a representative image from a number of MIT Indoor67 scene categories, along with the corresponding affine rectifications to the right of each detection.



(a)

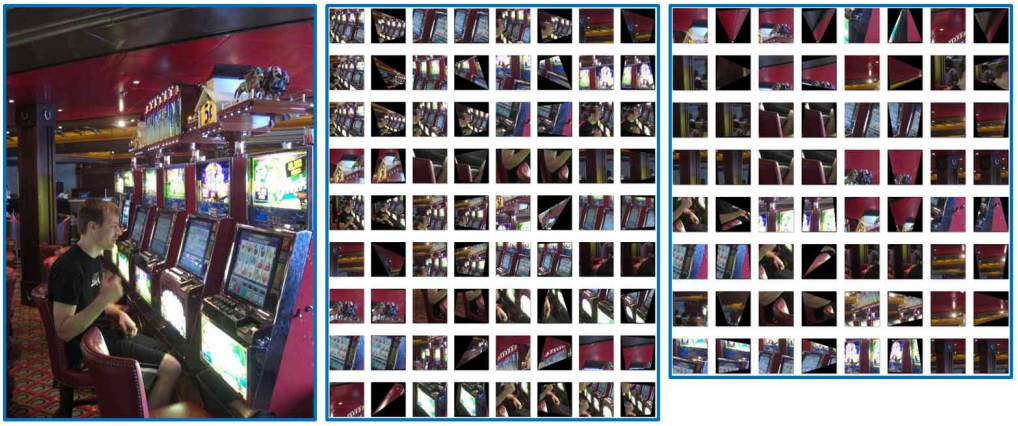


(b)



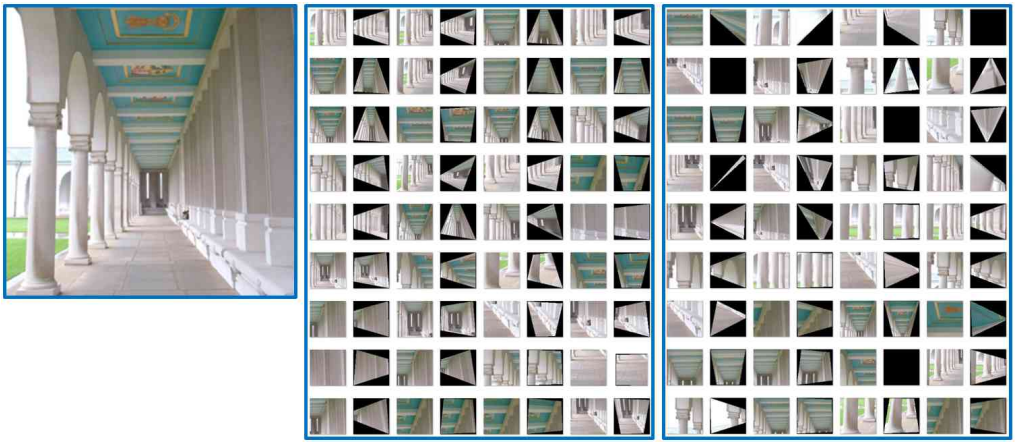
(c)

FIGURE 5.2: Detection of homogeneous texture: comparing PROPOSED method **(CENTER)** with TILT [149] **(RIGHT)**. Images **(LEFT)** sampled from (a) `airport_inside`, (b) `art_studio`, (c) `auditorium` — 1/3



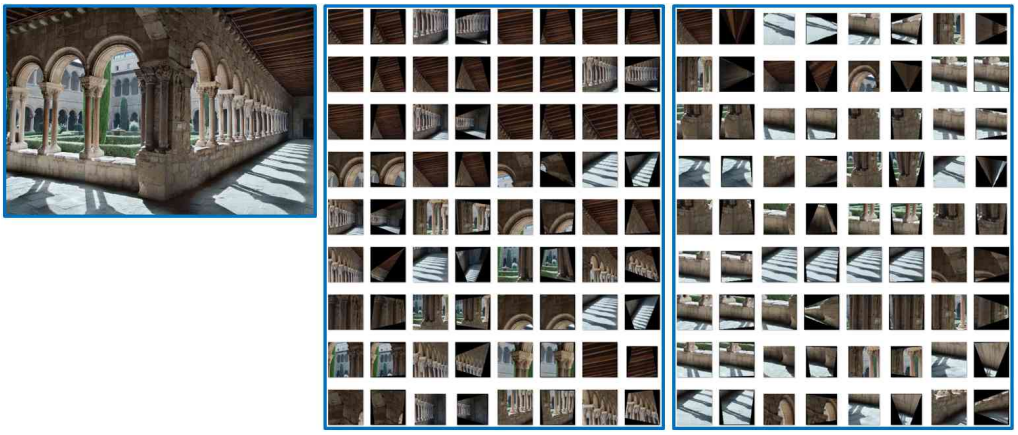
(d)

(e)

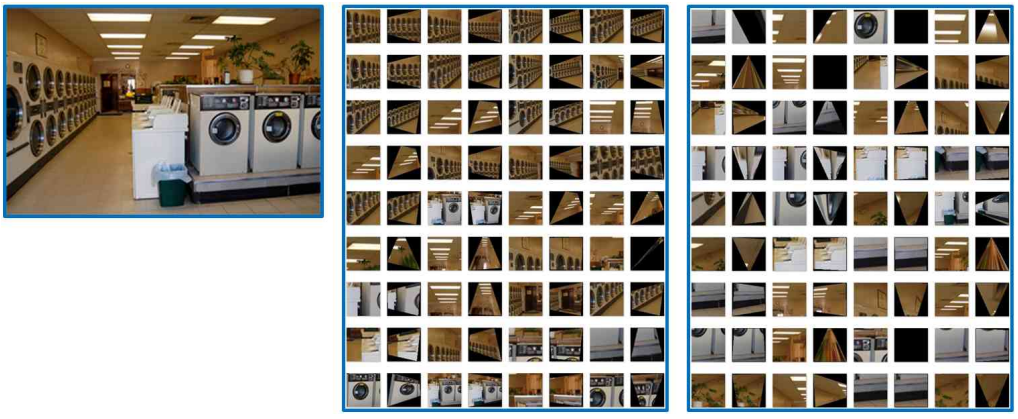


(f)

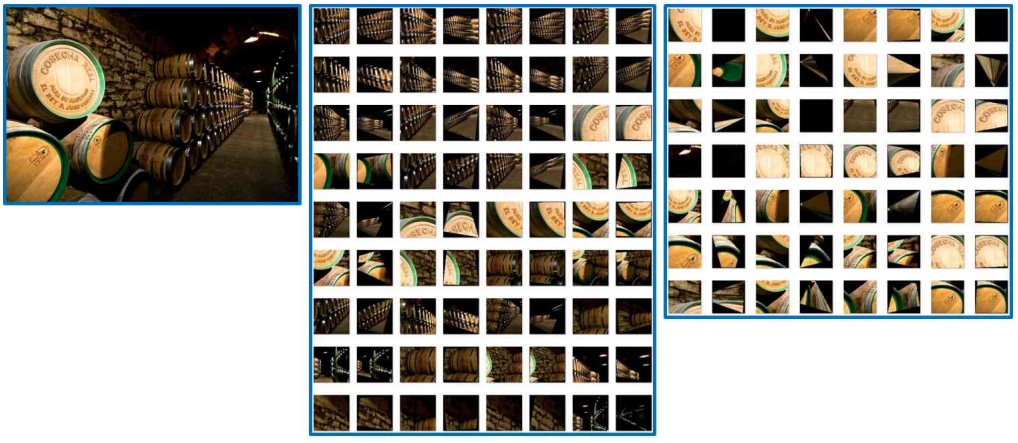
FIGURE 5.2: Detection of homogeneous texture: comparing PROPOSED method **(CENTER)** with TILT [149] **(RIGHT)**. Images **(LEFT)** sampled from (d) `casino`, (e) `classroom`, (f) `cloister` — 2/3



(g)

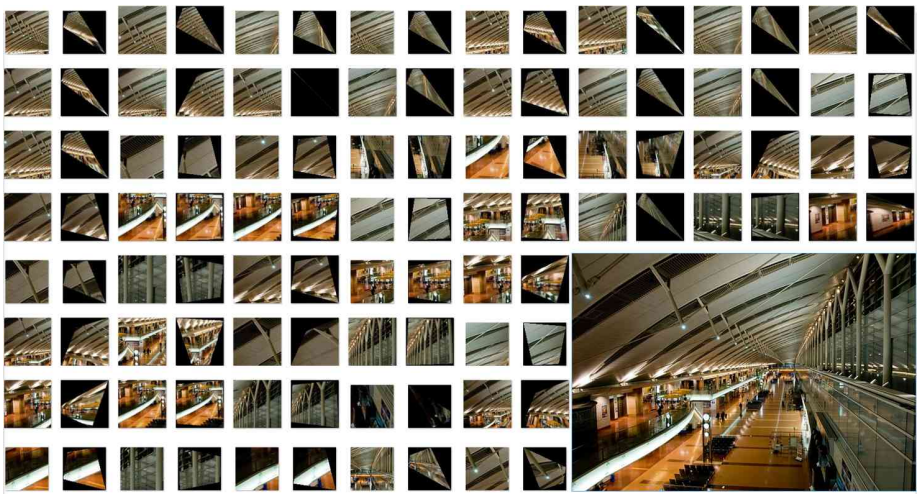


(h)

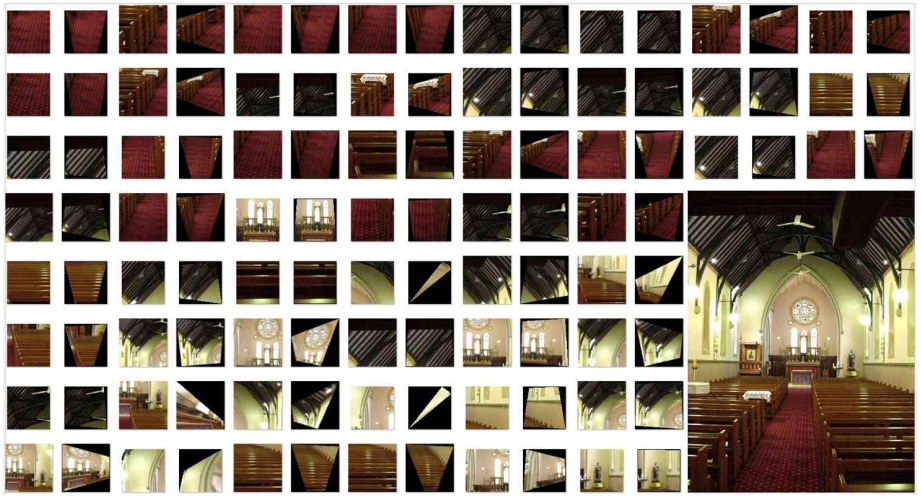


(i)

FIGURE 5.2: Detection of homogeneous texture: comparing PROPOSED method **(CENTER)** with TILT [149] **(RIGHT)**. Images **(LEFT)** sampled from (g) `cloister`, (h) `laundromat`, (i) `winecellar` — 3/3



(a)



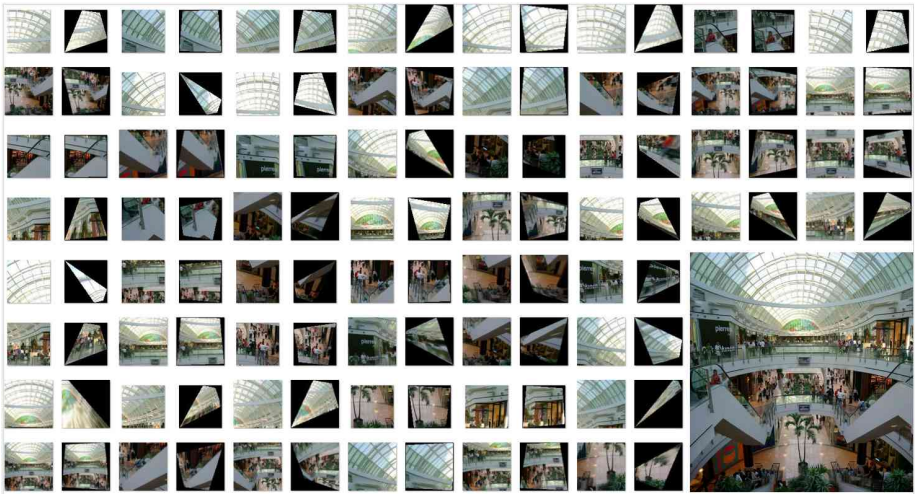
(b)

(c)

FIGURE 5.3: Detection of Homogeneous Texture by the proposed method. Images sampled from (a) `airport_inside`, (b) `church_inside`, (c) `concert_hall` — 1/5

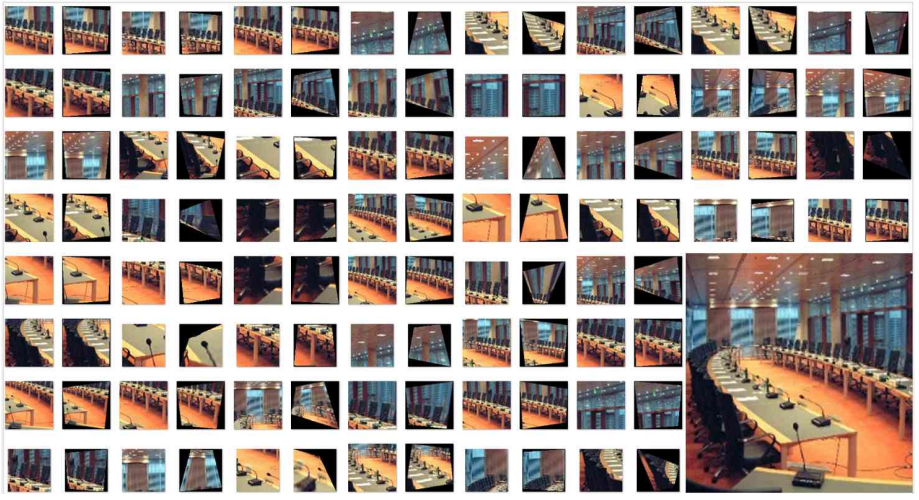
(d)

(e)



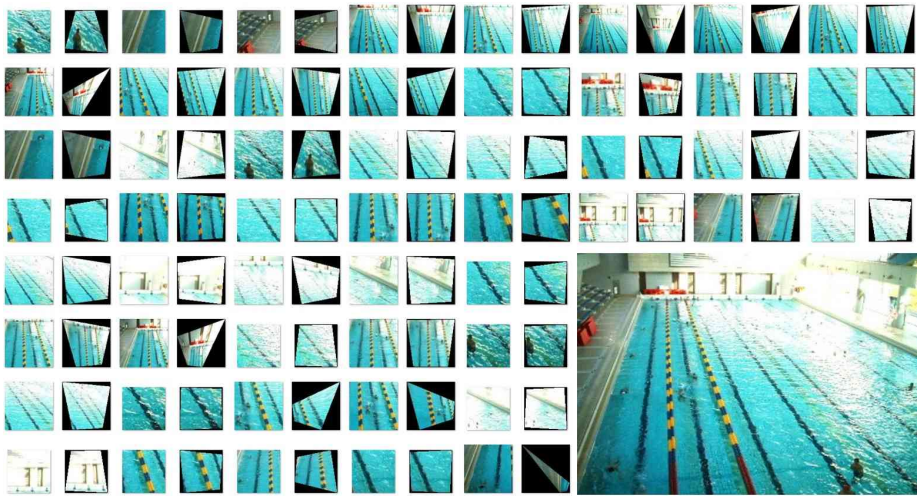
(f)

FIGURE 5.3: Detection of Homogeneous Texture by the proposed method. Images sampled from (d) `garage`, (e) `library`, (f) `mall` — 2/5



(g)

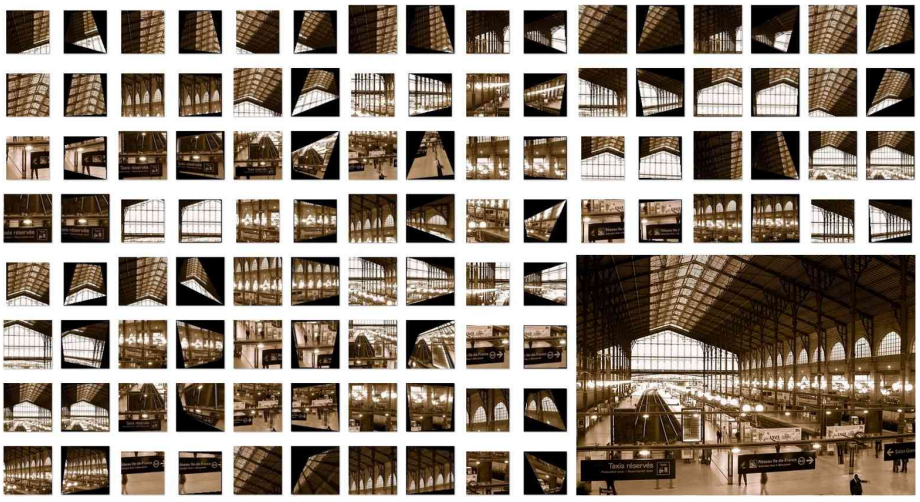
(h)



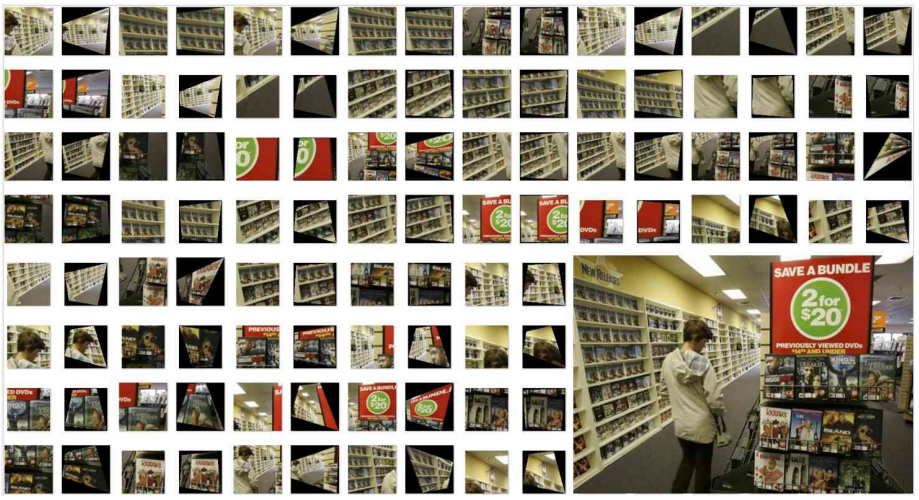
(i)

FIGURE 5.3: Detection of Homogeneous Texture by the proposed method. Images sampled from (g) `meeting_room`, (h) `movie_theater`, (i) `pool_inside` — 3/5

(j)



(k)



(l)

FIGURE 5.3: Detection of Homogeneous Texture by the proposed method. Images sampled from (j) `staircase`, (k) `trainstation`, (l) `video_store` — 4/5

(m)

(n)

(o)

FIGURE 5.3: Detection of Homogeneous Texture by the proposed method. Images sampled from (m) `warehouse`, (n) `winecellar`, (o) `winecellar` — 5/5

In general, it can be seen that TILT is able to localize meaningful texture only in a few cases (e.g., b), when the low-rank assumption is satisfied. Correct rectifications are usually obtained when a patch is free from outliers (e.g., some patches in f). By contrast, the proposed approach is seen to perform impressively in localizing and rectifying interesting homogeneous texture that can serve as meaningful mid-level scene features in all cases.

Fig. 5.3 presents additional qualitative results for the proposed scheme in representative images from various MIT Indoor67 scene categories. Photometric severities, such as significant illumination changes over a given texture [(i) `pool_inside`, (e) `library`], or poor lighting conditions [(o) `wine_cellar`] are unable to deter the algorithm. In cases with large clutter [(e) `mall`, (k) `trainstation`], the top-scoring patches tend to depict meaningful homogeneous texture. A remarkable resilience to outliers is seen — the frequency of repeating columns in (k) `train_station`, marred by sunlight beams, is appreciably recovered, while that of under-water pool lanes in (i) `pool_inside` is also accurately differentiated from the yellow tape above water.

Pertinent scales are localized in every case — e.g., coarse-scaled detections in (i) `pool_inside` and (i) `video_store`, as opposed to the fine-scaled detections on the textured flooring in (e) `library`. Patches at coarse scales tend to exhibit limited spatial support for the texture in question (see, e.g., (g) `meeting_room`, (n) `winecellar`), yet they can be reliably detected and correctly rectified. Also, the method can cover a wide range of frequencies — e.g., low-frequency texture in (i) `pool_inside` vs. high-frequency in (j) `staircase`.

The absence of long straight lines in the horizontal direction in (e) `library` and (j) `wine_cellar`, needed to obtain vanishing points for a scene layout estimation approach ([121, 51]), should be noted. Consequently, neither

can such a scheme localize the room surfaces, nor can it be relied upon to produce planar rectifications. As an aside, more examples of such texture lacking in straight lines appear in Figs. 4.2, 5.1. Further observe the presence of more than the usually-assumed three principal directions in (j) `staircase`, and (k) `train_station`, or the fact that room content is not always aligned with the principal directions ((g) `mall`). Finally, even in scenarios containing three principal directions, it may not be possible to reliably compute them. An example is Fig. 5.2(b) `cloister`, where the shadows can (and do) cause the estimation of vanishing points to fail. On the other hand, a local texture based approach to detection and rectification can be seen to successfully handle *all* the aforementioned problems.

A failure case of the proposed approach is when a patch upon rectification results in homogeneous texture, though it may not have a semantic meaning (at least, to humans). E.g., the top patch (undulating water) in Fig. 5.3(i) `pool_inside`. Another factor for failure was discussed in Sec. 4.6, i.e., the error measure upon which the number of outliers is determined is not affine invariant. While the heuristic normalization proposed therein has since been observed to perform favorably, occasional failures do occur.

5.3 Estimating Scene Spatial Layout

This section demonstrates an estimation of indoor scene layout (see Sec. 3.2) by **assigning a geometric class (left/right wall or ceiling/floor) to a homogeneous texture detection in a scene** (Sec. 5.2), and its recovered projective parameters. The vanishing line l of a plane Π passes through the two corresponding vanishing points, and is given by their cross product: $l = vp_1 \times vp_2$, where vp_1, vp_2 are 3-vectors specified in homogeneous coordinates. Fig. 5.4 depicts two planes Π_1, Π_2 that make up the

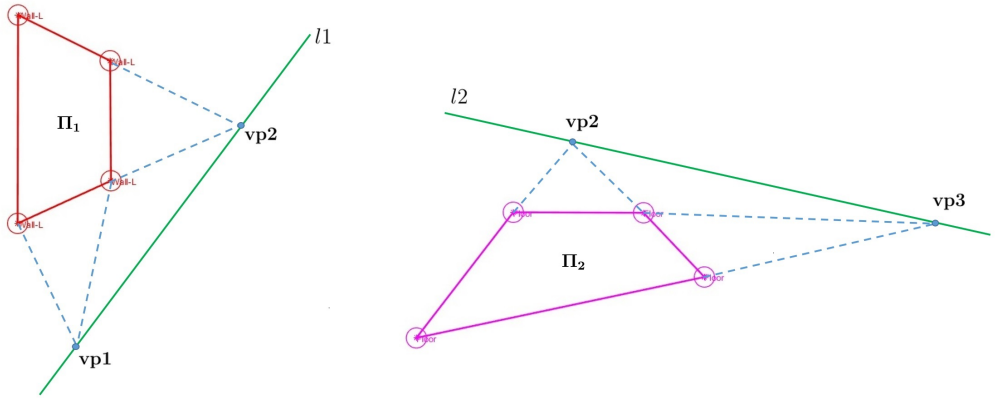


FIGURE 5.4: A scene plane may be classified as vertical or horizontal based on the slope of its vanishing line, if known, and as a left/right wall or ceiling/floor based on the position of this line w.r.t the plane. The vertical lines of plane Π_1 meet at infinity, but are shown to intersect at a finite point vp_1 for illustrating the vanishing line l_1.

left wall of a scene and its floor, respectively (vp_2 is the common vanishing point between the two planes). The wall, a vertical surface, tends to have a vanishing line l_1 that has a larger slope compared to that of the floor l_2, which is a horizontal surface. Now, the projective parameters $[h_7\ h_8\ 1]$ recovered for a given detection in fact happen to specify the vanishing line of the plane (see Sec. 2.7.2 in [47]) in the standard form $[a\ b\ c]$. The slope of the line:

$$\theta = \arctan\left(-\frac{h_7}{h_8}\right) \tag{5.1}$$

may be used to determine whether a detected homogeneous patch depicts a vertical surface or a horizontal surface (a fixed partition of 45° is used in experiments to separate horizontal and vertical planes). In addition, depending on the position of the line with respect to the patch center, the patch may be classified as left/right wall (if a vertically oriented vanishing line lies to the right/left of patch), or as ceiling/floor (if a horizontally oriented vanishing line lies below/above the patch). The horizontal position

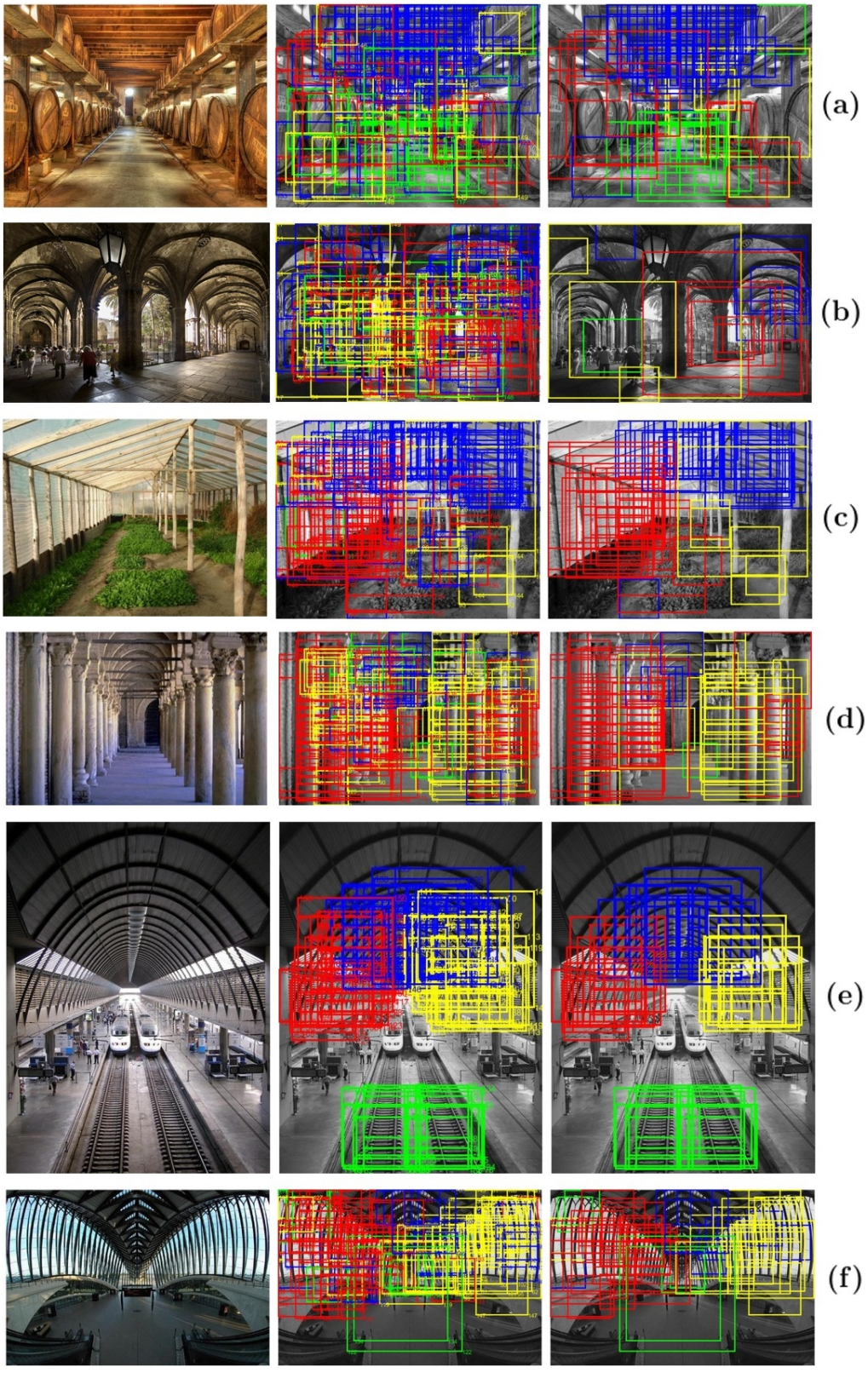


FIGURE 5.5: Scene layout estimation by homogeneous texture detections, and associated vanishing lines. Given scene (left), raw detections (center), post-NMS (right). Left wall = red, right wall = yellow, ceiling = blue, floor = green. For comparison with box layouts [51], c.f. Fig. 3.3. **Best viewed in color.**

of the line (needed to classify a patch as left/right vertical surface) is easily determined by computing the x-coordinate corresponding to the patch center's y-coordinate on the line, and vice versa if the vertical position of the line is required (needed to classify a patch as ceiling/floor).[1]

In principle, it is also possible to classify a given detection as frontal; if the vanishing line lies 'far' from the patch (based on some pre-defined threshold), it may be classified as a frontal surface exhibiting no or minute perspective distortion. Note the slope in this case is useless. In practice, however, it was observed that allowing for frontal planes caused misclassifications of planes as frontal that would otherwise be assigned to the vertical (walls) or horizontal (ceiling/floor) classes. This is since the recovery of projective parameters is not perfect — some times, only partial rectification is obtained. In other words, the algorithm thinks the perspective distortion in such cases is not pronounced, and consequently incorrectly labels these planes as frontal. This adversely increases false positives and decreases true positives.

Fig. 5.5 shows qualitative results obtained by the proposed approach on the same set of images as in Fig. 3.3. The first image in each row is the given indoor scene. The center image depicts the top 150 (based on proportion of RANSAC outliers) homogeneous detections in this scene. The box outlines are color-coded according to their geometric class as follows: left wall = red, right wall = yellow, ceiling = blue, floor = green. The figures in the third column are obtained by non-max suppression (NMS) performed *across* geometric classes, i.e., a **geometric class-aware NMS**. Specifically, the detections are ranked according to outlier score. Any incoming detection is *not* admitted if atleast 50% of its area is already occupied by *any* previously

[1]For a line in the general form $ax + by + c = 0$, the slope and y-intercept are given as $-a/b$ and $-c/b$, respectively. Thus, we have the slope of the vanishing line as $-h_7/h_8$ and the intercept as $-1/h_8$.

admitted patch, that is *not* from the same geometric class. Two detections of the same class do not suppress each other.

5.3.1 Comparison and Discussion

The planar structure in the scenes appearing in Fig. 5.5 satisfy the homogeneity assumption. Consequently, the proposed approach is easily able to overcome the challenges identified in Sec. 3.2.1, in particular, Fig. 3.3. Since multiple detections are used, a few incorrect detections are masked out by the correct ones (a). Forked layouts (b), angled ceilings (c), non-Manhattan structure (e,f), and textured multi-planar scenes, in general, can be naturally handled. Since the algorithm exploits *any* generic homogeneous texture, and not merely lines, their absence in any principal direction no longer poses a problem (d).

Fig. 5.6 presents some additional qualitative results from the proposed approach, as well as the box layout estimation method of Hedau. et. al. [51] for comparison. In what follows, interesting observations are made regarding the proposed scheme, and strengths and weaknesses of both methods are highlighted along the way.

In Figs. (a, b), the scenes largely lack homogeneous texture, except for that on the rug and bed, respectively. While spurious detections are obtained in the other regions, they are few. In Fig. (c) only the ceiling and one wall depict texture. On the other hand, [51] performs well on (a) and (c), but fails in (b) due to the angled ceiling. In (d, f), [51] fairs quite well, even successfully overcoming clutter (seating) in the auditorium scene, as it is trained to do. Misclassification of ceiling is observed in (e), however. The proposed method produces some mis-classifications, particularly in (d, e), but largely fairs well since homogeneous texture is abundantly present.

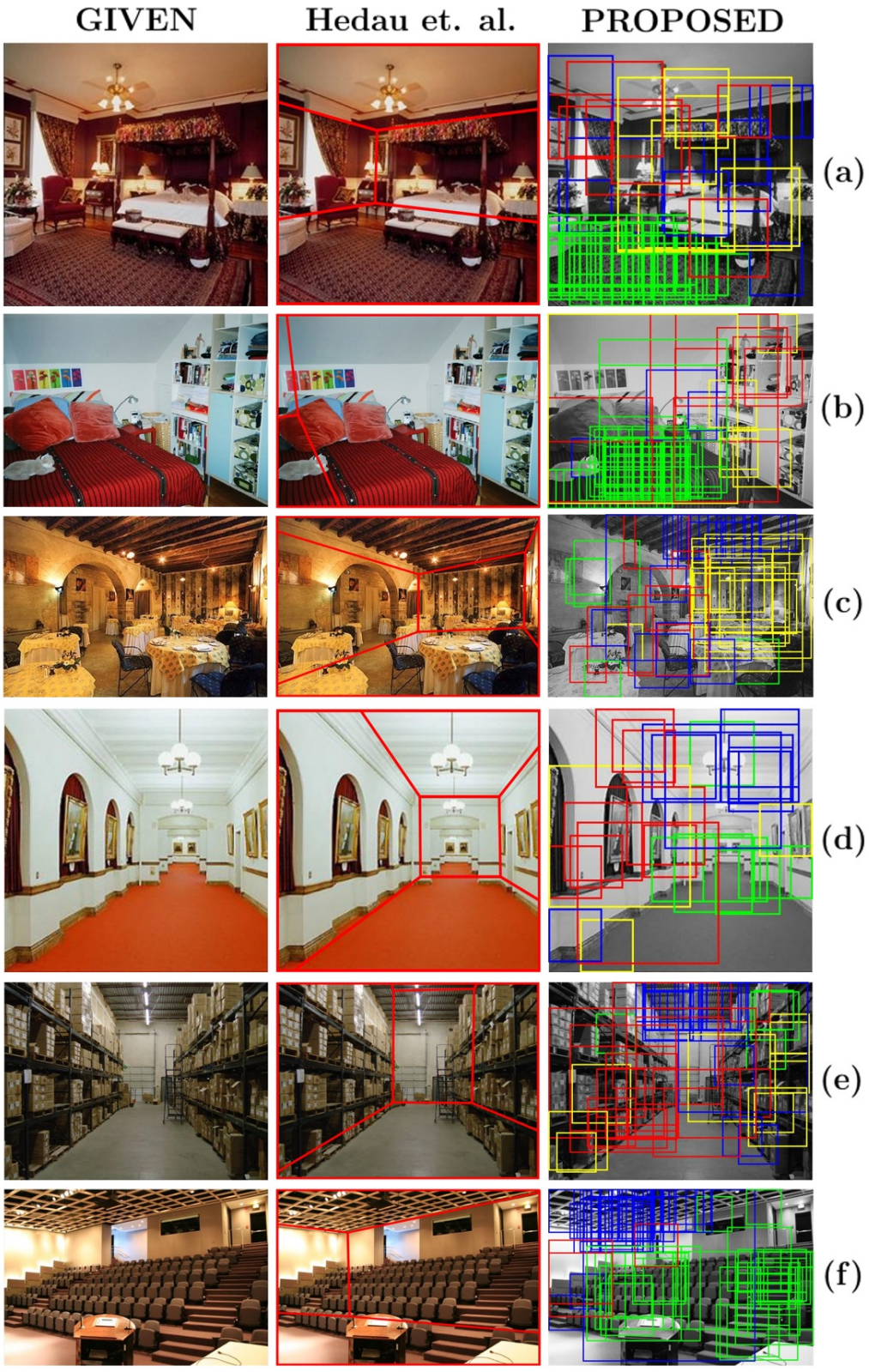


FIGURE 5.6: Qualitative comparison of box layout estimate [51] (center; using author implementation) with proposed method using homogeneous texture detections (right) — 1/2. Left wall = red, right wall = yellow, ceiling = blue, floor = green. **Best viewed in color.**

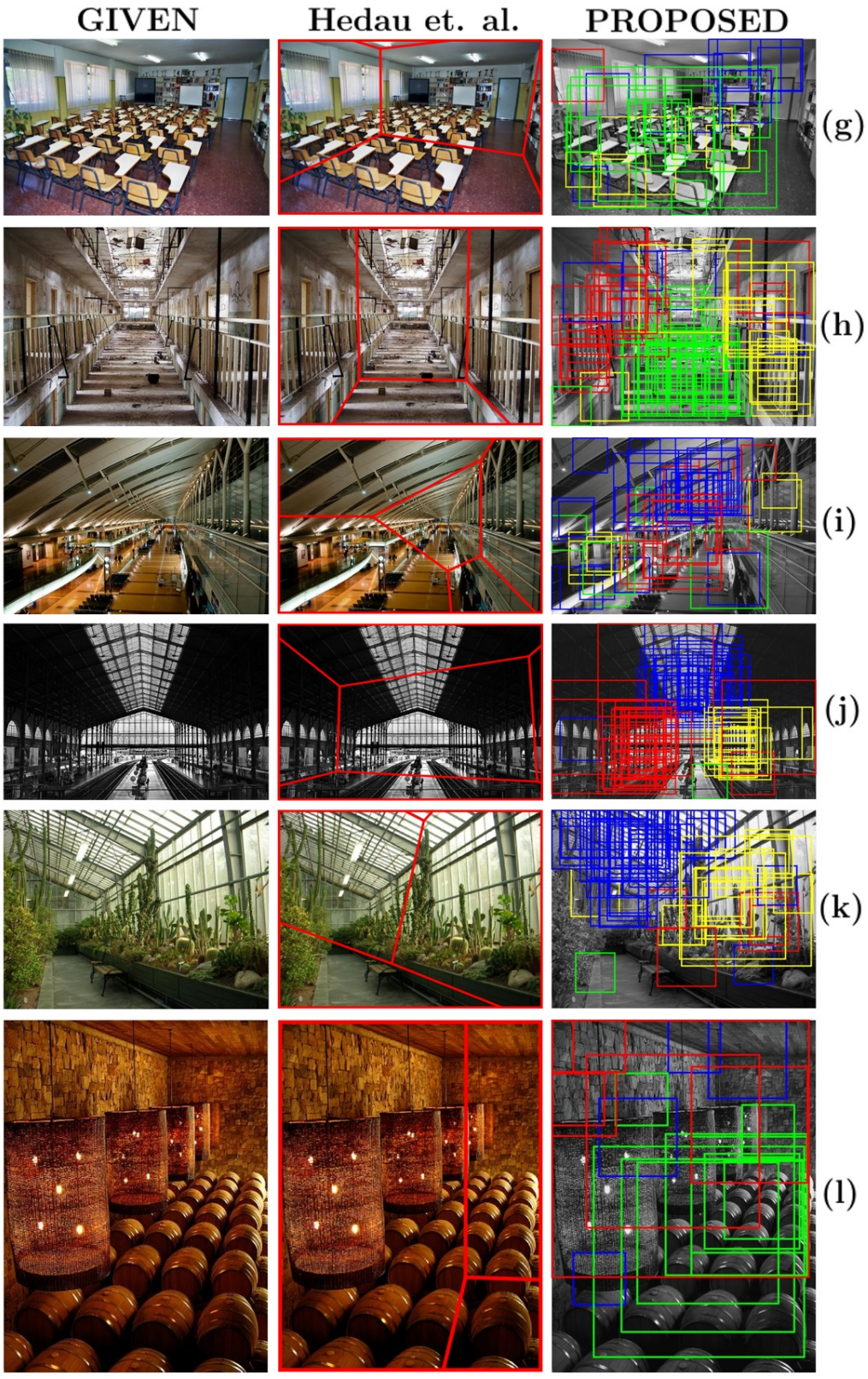


FIGURE 5.6: Qualitative comparison of box layout estimate [51] (center; using author implementation) with proposed method using homogeneous texture detections (right) — 2/2. Left wall = red, right wall = yellow, ceiling = blue, floor = green. **Best viewed in color.**

It successfully identifies the ceiling in (e), and, different from [51], assigns the floor category to seats in (f). This is because the oriented plane along which the seats recede has a vanishing line that is closer to the horizontal direction as opposed to the vertical direction.

Figs. (g – l) present cases where the proposed method performs better than [51], though (g, h, l) satisfy their box requirements. The multi-scale nature of detections from Sec. 5.2 is pronounced in (l), making it an interesting case.

Observe the proposed scheme uses neither vanishing points nor sophisticated machine learning with rich features sets to obtain a layout, yet can often do a better job than [51, 52] — provided the homogeneity assumption is satisfied in a given scene. However, our objective here is not to downplay the importance of previous work in this direction, but to draw attention of the community toward the potentials of shape from texture in such practical applications. Indeed, in a high-performance system aiming at obtaining scene surface layouts in any generic scene, machine learning would play an indispensable role.

One does observe some mis-classifications by the proposed scheme. This arises due to incorrect projective parameter estimation (either due to incorrect texture frequency estimate, or due to the non affine-invariance of our error measure), and hence incorrect estimate of the slope of a vanishing line. Currently, no spatial priors are enforced — a 'left wall' is just as likely to be detected on the right of a given image as on the left. It is possible to improve layout estimation by making use of a principled MRF formulation, however, that enforces priors such as ordering constraints [82]. However, not enforcing spatial priors allows flexibility, such as in the case of forked layouts [Fig. 5.5(b)]. Hence, an alternative possible post-processing mechanism to improve detections can be the modeling of semantic clusters

as Gaussians in space, and penalizing deviant detections. Moreover, the detections, and hence the layout estimations are likely to considerably improve should rectangular (as opposed to the squares currently used) patches be employed to capture texture elongated more in one direction than the other.

5.3.2 Non-Max Supression: A Tradeoff

So far we have observed and appreciated the pros of NMS. Let us now examine Fig. 5.7, which highlights the cons as well. Detections on the walls at coarse scales in (a, b) have suppressed those on the ceiling and/or floor — which are otherwise meaningful, valid true positives. In (c), detections firing on lateral views of barrels (red) have suppressed valid, top-view detections (green). (d) shows the case where detections on the floor tend to suppress those on the left and right vertical surfaces (grocery shelves). In (e), the red and yellow detections on the lateral views of church pews are valid. However, they are suppressed by the green detections, which are also valid and model a 'virtual' plane slanting away from the camera, parallel to the floor. In (f), blue, yellow and red detections correctly fire on the ceiling, right wall and left wall, but the red detections largely suppress the others. Due to this potential rejection of otherwise discriminative scene content, the classification experiments in Chapter 6 perform a different NMS, wherein patches only with the same geometric class, at the same image scale, and being sampled from the same anisotropic image representation are allowed to suppress each other (see Sec. 6.1).

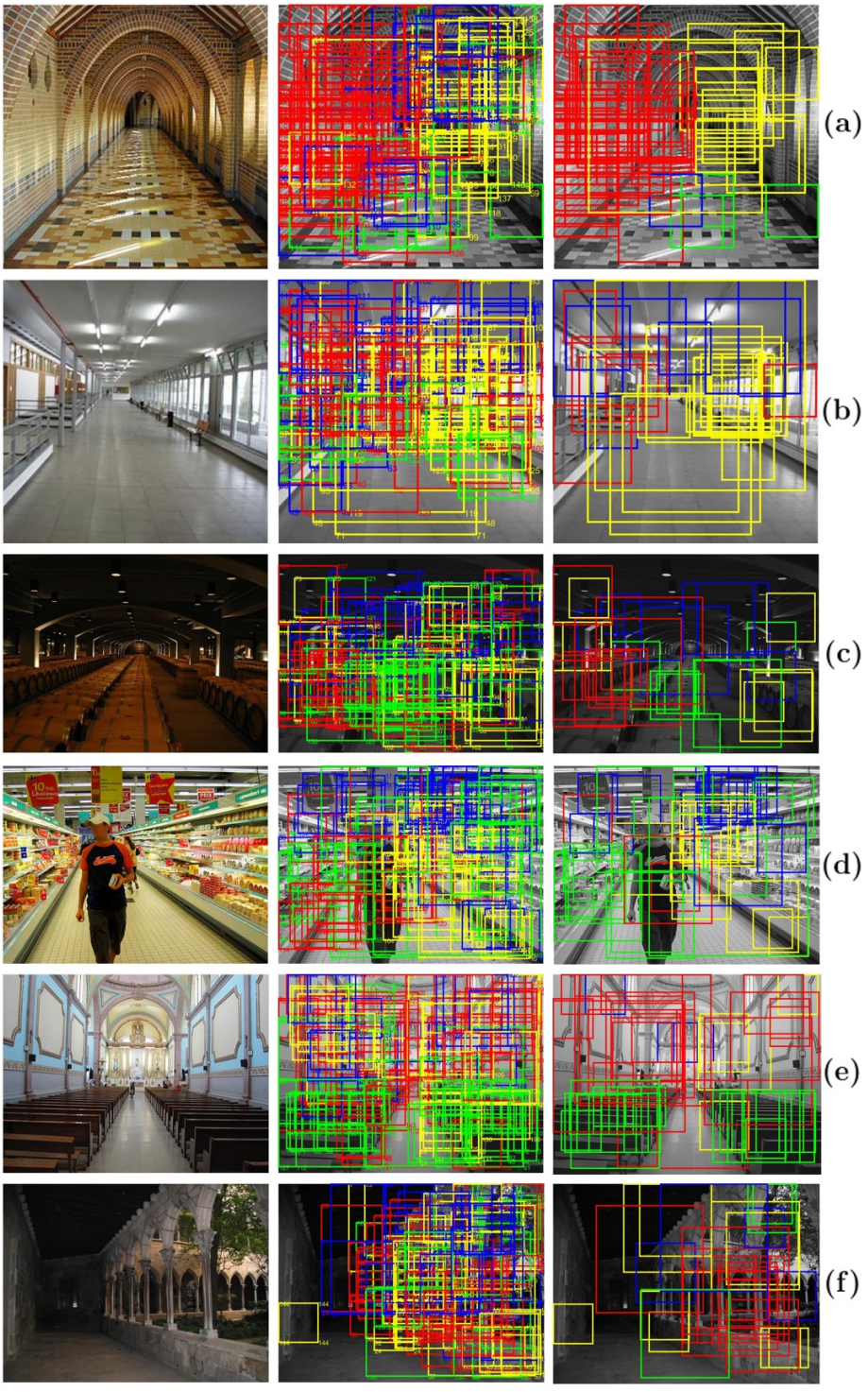


FIGURE 5.7: Inherent trade-off in enforcing non-max suppression when detecting homogeneous texture. Given scene (left), raw detections classified into geometric class (center), post-NMS (right). Coarse wall detections can suppress those on ceilings and floor (a - c), or vice versa (d - e). Conflict may arise between low walls and the backdrop (f). No NMS, however, can result in spurious detections (e). Left wall = red, right wall = yellow, ceiling = blue, floor = green. **Best viewed in color.**

5.4 Known Scene Vanishing Points Allow Metric Rectification

Scene vanishing points constrain the possible vanishing lines, and hence planes, manifested in the image. If the scene vanishing points in the image are known, or may be reliably computed, the corresponding vanishing lines (essentially, parameters h_7, h_8) may be obtained for each pair of vanishing points. For a *given* candidate pair of parameters h_7, h_8, Eqns. 4.9 and 4.11 reduce to linear least squares problems, where only u_s, v_s are to be computed. The pair of parameters minimizing the error 4.12 may be chosen as the winning candidate. Alternatively, having already estimated h_7, h_8 via non-linear least squares (Eqns. 4.9 and 4.11), the pair of vanishing points (among the candidates returned by some vanishing point detetion algorithm) that best satisfy the resulting estimated vanishing line may be chosen as the winning pair of vanishing points for the image patch in question. In other words, the process entails the use of robustly computed local texture cues to assign the correct pair of globally computed vanishing points to a textured surface in multi-planar scenes.

Known vanishing points can potentially correct any minor errors in rectification, and improve the detection of textured regions. Most importantly, it is possible to attain a rectification within only a scale ambiguity for a given patch if vanishing points are known. Here, we make the assumption that vanishig points as obtained for a patch are orthogonal.[2] Fig. 5.8 (left) shows a cloister scene, for which five line clusters were obtained (via a greedy clustering approach based on line-point voting), and two sample regions of homogeneous texture (green bounding box). For each of the 10

[2] A plane may exhibit vanishing points not in orthogonal directions; see Fig. 5.3(e) - `library` (the textured flooring) for an example

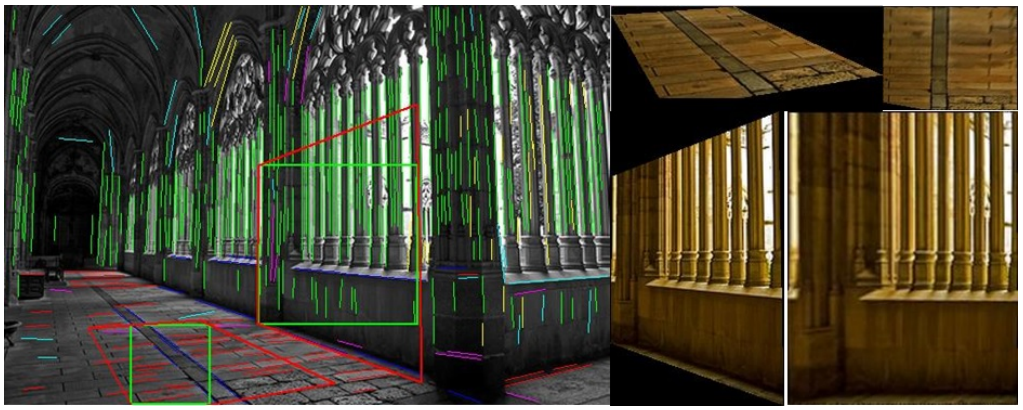


FIGURE 5.8: Using scene vanishing points in conjunction with homogeneous texture for metric rectification. Left: grayscale scene with overlayed line clusters assigned to vanishing points, and two sample regions of homogeneous texture (**Best viewed in color**). Right: sample regions cut out and rectified.

possible vanishing lines, Eqn. 4.11 was solved for the two patches, and the vanishing line $[h_7\ h_8\ 1]$ minimizing it was assigned to the patch. This may be used to obtain an affine rectification, which only restores parallelism. To also recover angles and any in-plane rotation, we proceed as follows. Now that the two vanishing points belonging to a patch are known, a circumscribed quadrilateral may be obtained (shown in red in Fig. 5.8) (left), such that the opposite edges intersect at the respective vanishing point. The vertices are ordered ABCD such that edge AB and CD form smaller angles with the horizontal compared to AD and BC, and AB is above CD. This definition of a canonical orientation of the quadrilateral is necessary to remove any in-plane rotation before feature extraction for recognition. A rectifying homography is now computed to warp the quadrilateral to a rectangle, restoring the orthogonality of the line directions (recall we have assumed the obtained vanishing points are from orthogonal directions). The results of this metric rectification are also shown in Fig. 5.8 (right).

In experiments for this thesis, however, estimating scene vanishing points

in the challenging MIT Indoor67 was not observed to be feasible. The problem of clustering lines based on membership to a dominant principal direction is essentially ill-posed. Moreover, severities such as room clutter, missing or few lines in a principal direction, non-conformance to the Manhattan assumption, and the frequent presence of more than three principal directions renders the task infeasible for current technology (see also Sec. 3.2.1). As such, the classification experiments in Chapter 6 make do with an affine-ambiguous rectification facilitated by the approach presented in Chapter 4.

5.5 Detection & Geometric Class Assignment: Quantitative Evaluation

In this section, a quantitative evaluation of the proposed detection (Sec. 5.2) and geometric class assignment (Sec. 5.3) is performed. It is based on a subset of 300 images sampled from the MIT Indoor67, with at least 3 from each scene category. This subset has been manually annotated with quadrilaterals indicating homogeneous textured regions, their plane projective parameters, and their geometric class IDs (left/right wall, ceiling, floor). Fig. 5.9(left) illustrates a sample annotated image.

Let us define **true positives** (TP), **false positives** (FP) and **false negatives** (FN) as follows.[3] For **precision** [TP/(TP+FP)], TP is the number of candidate patches whose estimated geometric class (Sec. 5.3) matches with an annotated region, with 50% intersection-over-detection (IOD, i.e.,

[3]Since our detector is not "trained" to produce an exact bounding box, we slightly differ in our definitions of these parameters from object detection [28]. Object detection methodology considers any more than one detection for a given ground truth as FPs, but all such detections are considered TPs in our scenario.

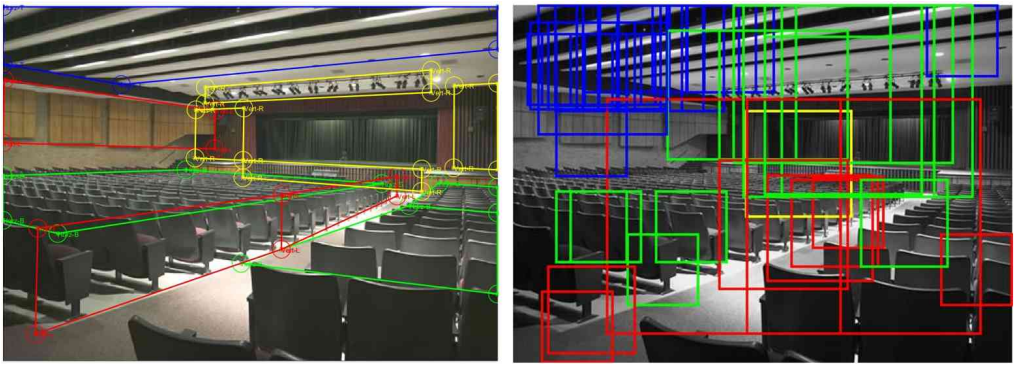
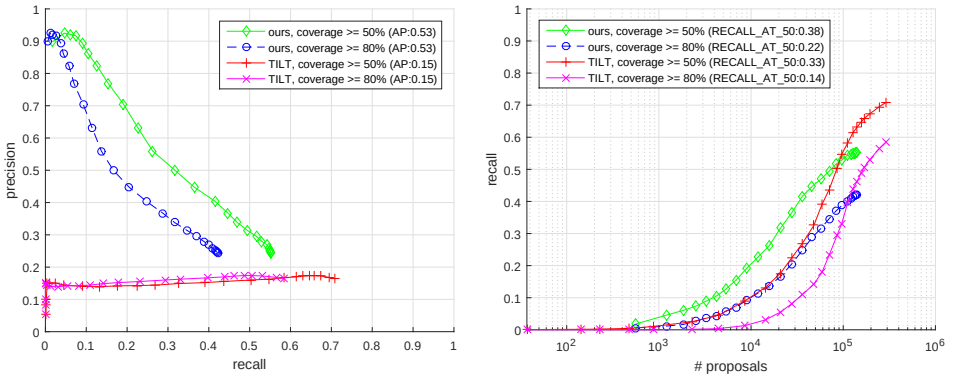


FIGURE 5.9: Annotation of indoor scene images to specify ground truth geometric class to a textured surface vs. the proposed method. **Left:** Images are annotated with quadrilaterals specifying left (red) / right (yellow) walls, ceiling (blue) and floor (green), using a custom GUI written for the purpose. **Right:** A geometric class ID is assigned to a detection based on its estimated vanishing line (Sec. 5.3), and a quantitative evaluation is performed based on precision and recall computed against the annotated ground truth. **Best viewed in color**.

at least 50% of the candidate's area should cover the annotation), while FP is a candidate that fails in this manner. For **recall** [TP/(TP+FN)], TP is the number of annotated regions that are "fired on" by one or more candidates (with the correct geometric class), such that its area beyond a certain threshold is covered (we evaluated at both coverage >= 50% and >= 80%), while FN is the number of annotated regions that fail in this manner. Note that for recall, TP + FN = 1367, which is the total number of annotated regions, similar to object detection[28].

Fig. 5.10 presents the precision-recall curves, and the recall vs. # proposals curves for our method, as well as for TILT [149] (for which the ratio of final to initial rank is used to obtain a decision score). One can observe a considerably more superior performance by our method, with an **average precision = 0.53**, compared to 0.15 by TILT. Both methods improve in recall with increasing #proposals, but the proposed approach is seen to maintain a larger recall for the same #proposals from the outset. This further corroborates the claim of this thesis in that existing tools to handle



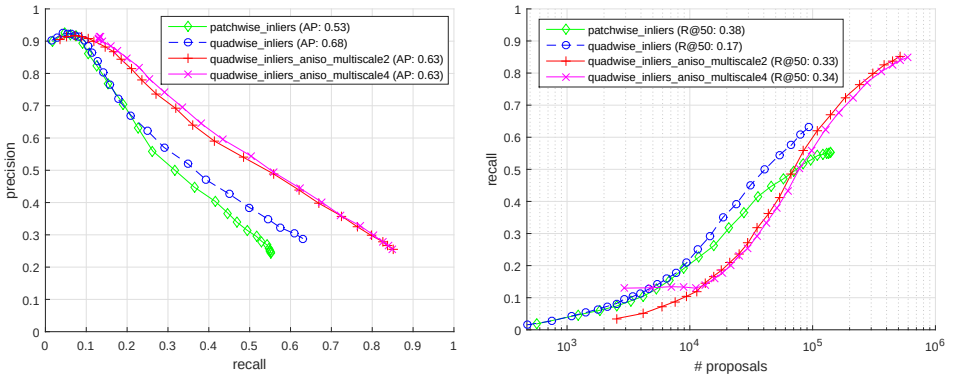
(a) (b)

FIGURE 5.10: (a) Precision-recall and (b) recall vs. # proposals curves for proposed detector and TILT [149].

METRIC / REP.	**proposed (coverage >= 0.5)**	**TILT (coverage >= 0.5)**	**proposed (coverage >= 0.8)**	**TILT (coverage >= 0.8)**
AP	**0.53**	0.15	0.53	0.15
AR	0.34	0.35	0.22	0.23
Precision 50$^{\text{th}}$ decision pt.	**0.44**	0.15	0.44	0.15
Recall 50$^{\text{th}}$ decision pt.	**0.38**	0.33	0.22	0.14
Precision 100$^{\text{th}}$ decision pt.	0.27	0.16	0.24	0.16
Recall 100$^{\text{th}}$ decision pt.	0.67	0.71	0.42	0.58

TABLE 5.1: Quantitative performance of proposed homogeneous texture detection vs. that by TILT [149].

texture in the wild are not up to par. Table 5.1 summarizes the average precision (AP) over the PR curve, and the average recall (AR) over the recall vs. # proposals curve, as well as details the precision and recall values at the 50^{th} and 100^{th} (i.e., using all proposals) decision points. Proposals with homogeneous image regions (low gradient energy) or those with a single recovered optimal dominant Gabor frequency but non-trivial projective parameters are discarded. Consequently, recall does not fully reach 1 in



(a) (b)

FIGURE 5.11: (a) Precision-recall and (b) recall vs. # proposals curves for proposed detector. Stricter decision scoring (requiring a certain % of inliers in all patch quadrants) improves AP. Additional anisotropic multiscale image representations introduce additional proposals, improving Recall.)

METRIC / REP.	**patch_wise**	**quad_wise**	**quad_wise aniso2**	**quad_wise aniso4**
AP	0.53	**0.68**	0.63	0.63
AR	0.34	0.25	**0.41**	**0.43**
Precision 50th decision pt.	0.44	0.75	0.68	0.68
Recall 50th decision pt.	0.38	0.17	0.33	0.34
Precision 100th decision pt.	0.27	0.27	0.24	0.24
Recall 100th decision pt.	0.67	0.67	**0.87**	**0.87**

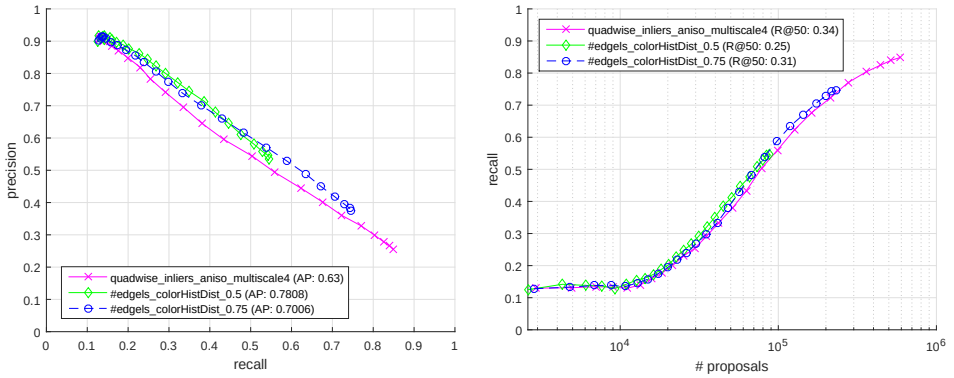
TABLE 5.2: Quantitative performance of various configurations of the proposed homogeneous texture detector. Stricter decision scoring (requiring a certain % inliers in all patch quadrants) improves AP. Additional anisotropic multi-scale image representations improve recall by introducing additional meaningful proposals. See text for details.

this evaluation.

An improved decision metric was also attempted, wherein the proportion of RANSAC inliers in all four patch quadrants (**quad_wise**) is used, instead of

over the entire patch (**patch_wise**). Intuitively, this implies that the inliers should be well-distributed over the patch, and not simply be concentrated in a limited region of it. Fig. 5.11 quantitatively demonstrates that this brings up the AP to 0.68, but the average recall (AR) falls considerably from 0.34 to 0.25. In a bid to increase this recall, additional anisotropic multi-scale image representations are employed, significantly improving recall rates for the proposed detector (summarized in Table 5.2). Specifically, **aniso2** introduces more proposals by using two additional image representations where either the # rows or columns are doubled. Similarly, **aniso4** further adds two more representations where either the # rows or columns are halved. This is similar in motivation to the anisotropic multi-scale image representations employed in Sec. 4.6 in wanting to make the scales of relevant image features more pertinent with respect to the size of the Gabor filters. This also effectively makes the patch size rectangular instead of square, which is often more representative of the homogeneous texture in real world scenes. This can be quantitatively observed from the recall rates which considerably improve when using aniso2 or aniso4.

Finally, color histogram consistency between neighbouring quadrants is enforced as a constraint to further improve precision. Specifically, 10-bin color histograms for each of the RGB channels are computed and concatenated in all quadrants. Neighbouring quadrants should exhibit similar histograms within a certain l_2 distance (we tried 0.5 and 0.75) for the patch to be considered as a proposal. Similarly, another constraint is used wherein at least three quadrants not possessing a certain proportion (12.5%) of the patch's total # edgels (edge pixels) are rejected. Fig. 5.12 and Table 5.3 show these constraints can considerably improve precision, though at the cost of some drop in recall. It should be noted that the quantitative evaluation presented here can be considered as that for both the tasks of detection as



(a) (b)

FIGURE 5.12: (a) Precision-recall and (b) recall vs. # proposals curves for proposed detector. Pushing AP further at the cost of recall by imposing pre-filtering heuristics / constraints.

METRIC / REP.	**quad_wise aniso4**	**edgels_ colorHist_0.5**	**edgels_ colorHist_0.75**
AP	0.63	**0.78**	**0.70**
AR	0.43	0.30	0.38
Precision 50th decision pt.	0.68	0.84	0.76
Recall 50th decision pt.	0.34	0.25	0.31
Precision 75th decision pt.	0.38	0.66	**0.5080**
Recall 75th decision pt.	0.70	0.43	**0.6147**
Precision 100th decision pt.	0.24	0.53	0.37
Recall 100th decision pt.	0.87	0.55	0.75

TABLE 5.3: Quantitative performance of various configurations of the proposed homogeneous texture detector. AP may be pushed further, at the cost of recall, by imposing pre-filtering heuristics requiring consistency of color histograms and edgels in all patch quadrants.

well as geometric class assignment. This is since it is really the assignment (via proposed approach) of a geometric class to a given proposal that goes on to determine the detector's precision and recall.

Chapter 6

Indoor Scene Classification via Affine-Rectified Homogeneous Texture

Having robustly detected characteristic homogeneous texture in indoor scenes, can they be exploited for the purpose of scene semantic classification? This chapter aims to answer this question by performing a comprehensive set of classification experiments on the benchmark MIT Indoor 67-category dataset, spanning 6700 images ([106], Sec. 2.10). Sec. 6.1 discloses the approach and implementation details of the classification pipeline. Both regular (i.e., without any homogeneous texture detection or rectification) and rectified (i.e., based on detection and affine rectification) features are extracted. Four types of hand-crafted local texture descriptors have been employed: the thresholding based CENTRIST and LBP, as well as the gradient based SIFT and HOG. As a fifth descriptor, deep CNN features are also experimented with. Sec. 6.2 discusses the results in detail. To more rigorously evaluate the thesis, Sec. 6.3 applies the approach to an

additional 6200 images spanning 31 scene categories, being a subset of the ILSVRC 2015's Places2 dataset ([150], Sec. 2.10), consisting of 1 natural, 5 indoor and the remaining 25 being man-made outdoor environments, all of which tend to exhibit regular, repeating structure. The results demonstrate that rectification based on texture cues yields class-discriminative features that are also complementary to regular features.

6.1 Implementation Details

In what follows, a configuration is described for extracting descriptors and performing classification, and is common across all experiments. All images are resized to a reference scale, such that the smaller dimension is 400 pixels, and the aspect ratio preserved.

Feature Extraction — Regular Rep. For regular features (no rectification), patches sized 16x16 pixels are extracted on a regular grid with a spatial stride of 8 pixels (4 pixels for SIFT), with the reference image represented at the *same* set of 9 scales as determined in Sec. 5.2.1.

Feature Extraction — Rectified Rep. For a rectified representation, an 80x80 pixel detection is warped (using bilinear interpolation for speed) to a fixed size of 80x80 pixels. Then, patches sized 16x16 pixels are extracted from this warped region on a regular grid with a spatial stride of 8 pixels (4 pixels for SIFT). A patch in the warped region not fully visible in the original non-rectified region is rejected. A single scale is used (i.e., scale = 1), thereby retaining the scale at which a homogeneous textured region is detected. For learning the dictionary, the detector at the "edgels_colorHist_0.75" at the 75^{th} decision point is used (Table 5.3). This provides a good tradeoff between precision (50.8%) and recall (61.47%).

However, for training the kernels for classification, we rectify all available 80x80 pixel patches (after NMS), but discard those with homogeneous image regions (low gradient energy) or those with a single recovered optimal dominant Gabor frequency but non-trivial projective parameters (i.e., configuration "quad_wise aniso4, 100^{th} decision point" in Table 5.2). This configuration, though not so precise (0.24), affords a very high recall (0.87). For both cases (dictionary learning or kernel training), an **intra-geometric class, intra-scale and intra-aspect non-max suppression** is performed. Specifically, the detections are ranked according to outlier score. Any incoming detection is *not* admitted if at least 50% of its area is already occupied by *any* previously admitted patch, that is also 1) from the same geometric class, 2) at the same image scale, and 3) sampled from the same anisotropic image representation. Note this NMS for classification differs from that employed for layout estimation as described in Sec. 5.3, for reasons discussed in Sec. 5.3.2. Though it lends to a somewhat sparser image representation, potentially causing some loss in discriminative power, NMS is necessary to keep the computational requirements feasible.

Feature Encoding. Best practices for dense local feature based classification, as suggested in [11, 63] are followed. Specifically, the descriptor dimensionality is reduced to 80 features via PCA (except $LBP_{8,1}$, which is already 59-dimensional to begin with), followed by learning a 256-component GMM. Separate dictionaries for regular and rectified features are learned, using a sample of 10^6 features, obtained equally over the entire training set. A 2-level spatial pyramid (see [70], Sec. 2.3.2) is constructed, wherein a Fisher Encoding with sum pooling [128] is performed over each of the 5 spatial bins, obtaining a 40,960-dimensional descriptor per bin. Different from [11] (who normalize each bin separately), descriptors at each *level* of the spatial pyramid are l_2-normalized separately (i.e., 1 at the first level

and the concatenated 4 at the second level), since this was observed to give a better performance. Hellinger kernel mapping is then performed on the descriptors, followed by an l_2-normalization (as before) again, thereby obtaining a so-called Improved Fisher Vector (IFV). The 5 descriptors are then concatenated to obtain a 204,800-dimensional image representation. Classification is performed by linear (having already incorporated a non-linear Hellinger mapping) one-vs-all SVMs, using the code made available by [10].

Classification. Classification performance is reported as an average of 3 runs using the standard train-test split for the MIT Indoor 67 [106] (Sec. 6.2) (with the difference in each run being sampling of a subset of descriptors for dictionary learning, which is randomly performed). As is standard practice on this dataset, classification accuracy is defined as the average of the diagonal of the confusion matrix (i.e., average of per-class rates rather than average over all dataset). The same approach is taken for the subset of the Places2 dataset (Sec. 6.3). For obtaining the classification performance of a combined representation, soft-max transformed SVM scores of individual representations are multiplied, as proposed in [95] (and reviewed in Sec. 2.6).

6.2 Experiments on the MIT Indoor67 [106]

6.2.1 CENTRIST Descriptors

Table 6.1 presents the performance when using CENTRIST descriptors (see [137], Sec. 2.7). Reducing descriptor dimensionality (originally at 256, l_1 normalized to 1) to 80 via PCA was observed to give better classification performance with Fisher encoding, as opposed to 40 dimensions (done by

the original work on CENTRIST [137], though with a bag-of-words encoding). As an aside, an inhouse, multiscale implementation of sPACT (regular features; no detection or rectification) — CEN_SBOW — achieved 42.64% (HIK) compared to the 36.88% (RBF) reported in [137]. However, Fisher encoding (CEN) gives an even higher performance.

The slight loss in performance with a rectified representation (CEN_Rect) as compared to a regular representation (CEN) is likely because in the case of the former, the dictionary learned is essentially representative of only rectified homogeneous texture, which, although abundant, is still manifested at sparse locations. Nevertheless a rectified representation is still highly discriminative. More interestingly, both the regular and rectified features are highly complementary to each other, significantly boosting performance when used together (CEN + CEN_Rect).

Single Rep.	**% Accuracy**
CEN_SBOW [137]	36.88%
CEN_SBOW	42.64%
CEN	$46.44 \pm 0.62\%$
CEN_Rect	$45.36 \pm 0.36\%$
Combined Rep.	**% Accuracy**
CEN + CEN_Rect	$\mathbf{49.68 \pm 0.11}\%$

TABLE 6.1: MIT Indoor67 classification performance improvement with dense feature description of affine-rectified texture — CENTRIST.

6.2.2 LBP Descriptors

Local Binary Patterns (LBP) [93] is a discriminative texture operator, invariant to monotonic gray scale transformations, and widely employed in applications such as material and face classification. It thresholds the local neighbourhood at the gray value of the center pixel, and sums the

resulting bits over the neighbourhood, weighted by powers of 2. A histogram descriptor may then be computed over the resulting binary code image. A modification, LBP^{ri}, achieves invariance to rotation of local pixel neighbourhood by circularly rotating a binary code into its minimum value. Another enhancement is LBP^{u2}, where the histogram assigns all 'non-uniform' patterns to a single bin but maintains a separate bin for each 'uniform' pattern. A pattern is called uniform if it contains at most two bitwise transitions from 0 to 1, or vice versa. This variant reduces the descriptor size yet gives it more discriminative power. Yet another modification, invariant to *global* rotations, is proposed in [2]. Named *LBP-HF* (histogram of Fourier Features), it exploits a property of DFT whereby a cyclic shift in the input sequence (a histogram based on LBP^{u2}) causes a phase shift in the DFT coefficients. The *LBP-HF* outperforms the rotation-sensitive LBP^{u2}, as well as the locally rotation invariant LBP^{ri} on texture classification tasks [2]. Depending on the configuration used, and the circular neighbourhood parameters (P, R) (P = # pixels, R = radius), the length of the resulting histogram — essentially the image descriptor — varies.

Xiao et. al. [139, 138] have previously employed LBP features for scene classification on their 397-category SUN dataset, reporting a rather low performance of 14.7% by LBP^{u2} and 10.9% by *LBP-HF*, suggesting that incorporating rotation invariance is detrimental for scene recognition. One notes, however, that they have followed an approach similar to texture classification, computing one LBP descriptor per scene image, with the Histogram Intersection Kernel for classification. It is very likely, therefore, that the discriminative power of LBP features may have been downplayed (as opposed to, e.g., Dense SIFT at 23.5% or Dense HOG2x2 at 26.3%) in their evaluation.

Rep.	**% Accuracy (8,1)**	**% Accuracy (16,2)**	**% Accuracy (24,3)**
LBP_u2	$43.63 \pm 0.50\%$	$42.51 \pm 0.38\%$	$36.72 \pm 0.39\%$
LBP_u2_Rect	$42.34 \pm 0.42\%$	$\mathbf{44.30 \pm 0.34}\%$	$\mathbf{40.39 \pm 0.40}\%$
LBP_u2 + LBP_u2_Rect	$\mathbf{46.47 \pm 0.45}\%$	$\mathbf{45.72 \pm 0.28}\%$	$\mathbf{41.65 \pm 0.62}\%$

TABLE 6.2: MIT Indoor67 classification performance improvement with dense feature description of affine-rectified texture — Local Binary Patterns LBP^{u2}.

On the other hand, the experiments reported in this section extract LBP^{u2} and *LBP-HF* descriptors for densely sampled, overlapping patches. Three neighbourhood configurations (P, R) are used — (8,1), (16,2) and (24,3), yielding 59-, 243-, and 555-dimensional descriptors, which are then l_1 normalized to 1. The dimensionality for the last two cases was reduced to 80 features, before learning a GMM and performing Fisher encoding. The patch extraction, encoding and classification parameters are as described in Sec. 6.1.

Tables 6.2 and 6.3 present the MIT Indoor67 classification results. Interestingly, both the non-rotation invariant LBP^{u2} and the globally rotation invariant *LBP-HF* perform almost the same. The powerful Fisher encoding scheme seems to make up for the sensitivity of LBP^{u2} to rotation,

Rep.	**% Accuracy (8,1)**	**% Accuracy (16,2)**	**% Accuracy (24,3)**
LBP_HF	$44.02 \pm 0.51\%$	$42.88 \pm 0.05\%$	$36.85 \pm 0.84\%$
LBP_HF_Rect	$42.65 \pm 0.32\%$	$\mathbf{43.76 \pm 0.23}\%$	$\mathbf{40.47 \pm 0.58}\%$
LBP_HF + LBP_HF_Rect	$\mathbf{46.59 \pm 0.43}\%$	$\mathbf{46.26 \pm 0.40}\%$	$\mathbf{41.61 \pm 0.51}\%$

TABLE 6.3: MIT Indoor67 classification performance improvement with dense feature description of affine-rectified texture — Local Binary Patterns *LBP-HF*.

while contrary to [139, 138] incorporating rotation invariance (*LBP-HF*) is not detrimental to scene recognition. What is even more interesting, LBP descriptors extracted upon affine-rectification perform substantially better than regular descriptors for $(P, R) = (16, 2), (24, 3)$. Furthermore, as with CENTRIST descriptors, the rectified representations are not only class-discriminative, but also complementary to regular representations.

The drop in performance for the configuration (24,3) is likely because although the LBP coded image was constructed based on a 3-pixel radius neighbourhood containing 24 points around the center pixel, the final histogram descriptor for each patch was still obtained over a 16x16 pixel patch (to be consistent with the settings for the remaining descriptors (CENTRIST, SIFT and HOG2x2) in our evaluation). Moreover, the general trend in performance drop as the radius increases is because the LBP image construction causes an image border equal in size to the radius being discarded, thereby reducing features.

6.2.3 SIFT Descriptors

[63] have previously reported a performance of 60.77% on the MIT Indoor67 using RootSIFT descriptors (though, they use somewhat different patch and scale parameters than used here, a different SVM solver, and careful parameter cross-validation). Indeed, experiments with original SIFT (Sec. 2.3.1) yielded a lower performance of 59.14%. Therefore, following [63], this section reports results obtained with RootSIFT descriptors. RootSIFT is simply an element-wise square root of the l_1 normalized SIFT descriptors,

Rep.	**% Accuracy**
SIFT	$60.93 \pm 0.60\%$
SIFT_Rect	$60.88 \pm 0.32\%$
SIFT + SIFT_Rect	$\mathbf{63.01 \pm 0.19}\%$

TABLE 6.4: MIT Indoor67 classification performance improvement with dense feature description of affine-rectified texture — SIFT.

and evaluating Euclidean distances between RootSIFT vectors is essentially equivalent to using Hellinger kernel on original SIFT [4].[1]

As seen in Table 6.4, a dense representation based on rectified homogeneous texture (SIFT_Rect) performs almost the same as the regular representation. In other words, a rectified representation is just as highly discriminative. As with CENTRIST and LBP, it is also strongly complementary to a regular representation.

6.2.4 HOG2x2 Descriptors

The fourth set of experiments uses the HOG2x2 descriptor (see Sec. 2.3.1). Table 6.5 presents the results. The regular and rectified HOG perform lower compared to SIFT (Table 6.4). This finding is the opposite of that reported

[1] Incidentally, it is to be compatible with [63] that a denser grid spacing of 4 pixels is used in our experiments for SIFT feature extraction (though computationally expensive for rectified representation), while the other three descriptors use 8 pixels.

Rep.	**% Accuracy**
HOG	$57.69 \pm 0.30\%$
HOG_Rect	$\mathbf{59.70 \pm 0.39}\%$
HOG + HOG_Rect	$\mathbf{62.05 \pm 0.10}\%$

TABLE 6.5: MIT Indoor67 classification performance improvement with dense feature description of affine-rectified texture — HOG.

Combined Rep.	**% Accuracy**
SIFT + HOG	$62.30 \pm 0.52\%$
SIFT_Rect + HOG_Rect	$\mathbf{62.66 \pm 0.21}\%$
SIFT + HOG + SIFT_Rect + HOG_Rect	$\mathbf{64.56 \pm 0.02}\%$

TABLE 6.6: MIT Indoor67 classification performance improvement with dense feature description of affine-rectified texture — SIFT and HOG.

in [139, 138] for the SUN397 dataset, since their experiments employ a very different set of parameters.

Interestingly, not only is rectified HOG significantly more discriminative compared to regular HOG, but the two are strongly complementary, as with the previous three features. The high performance of rectified SIFT and HOG, as opposed to the thresholding based CENTRIST and LBP, is likely because these descriptors are essentially histograms of oriented gradients. As motivated in Sec. 4.1, rectification aligns features to a canonical coordinate frame, mitigating intra-class variations due to perspective effects or viewpoint differences, thereby facilitating recognition.

Table 6.6 reports classification results based on combining SIFT and HOG scores. By virtue of rectification via homogeneous texture cues, this thesis is able to achieve a performance of **64.54%** on the benchmark MIT Indoor67. This compares favorably with state-of-the-art approaches based on combined representations [Tables 2.1(top half) and 2.2(top half)], especially considering that *all* of them (except SIFT) employ learning based approaches to extract features. Additionally, ISPR is particularly trained to minimize classification error, while OPM makes use of an additional dataset to learn to determine planar orientations. In contrast the approach taken in this thesis does not involve any learning during feature extraction.

6.2.5 Deep ConvNet Descriptors

As reviewed in Sec. 2.9, pre-trained deep features learned via multi-layered ConvNets on huge datasets for the task of large scale object recognition have been successfully applied to other domains as off-the-shelf descriptors, including scene recognition. Where training a deep CNN requires special hardware and days or weeks to train, obtaining descriptors based on a pre-trained model for classification can be done in a matter of a few hours. The experiments here make use of the MatConvNet toolbox [129] and the pre-trained 16-layered **VGG-VD** CNN model [118] to extract deep CNN descriptors.

Regular Rep. The VGG-VD CNN requires the image be resized to a fixed size of 224x224 pixels, and a pre-learned "average image" be subtracted from it. In the process, aspect ratio is not preserved. A single, 4096-dimensional descriptor for the image is then obtained by using the output of the first fully-connected layer (specifically, layer# 14), and l_2 normalized to 1.

Rectified Rep. The detections obtained at the configuration "edgels_colorHist_0.75" at the 75^{th} decision point are used (Table 5.3), with precision 50.8% and recall 61.47%. NMS as described in Sec. 6.1 is performed. Each 80x80 pixel detection is warped (also to a size of 80x80 pixels) based on its recovered projective parameters, and a single 4096-dimensional descriptor is obtained from it as described above. All resulting descriptors are then subjected to an element-wise max or sum operation to obtain a single descriptor for the given image, which is then l_2 normalized to 1.

Having obtained image descriptors for both the regular and rectified representations, a linear SVM is used for classification. Table 6.7 presents the

Rep.	**% Accuracy**
CNN	68.57%
CNN_Rect(sum)	58.81%
CNN_Rect(max)	60.95%
CNN + CNN_Rect(sum)	**70.30**%
CNN + CNN_Rect(max)	**73.52**%

TABLE 6.7: MIT Indoor67 classification performance improvement with off-the-shelf deep CNN feature description of affine-rectified texture.

results. We observe that an accuracy of 68.57%[2] obtained by CNN image description is very impressive, especially since the dimensionality is merely 4096 and a linear kernel SVM is used. By contrast, a Fisher encoding descriptor, as used in Sec. 6.2.1, 6.2.2, 6.2.3, 6.2.4 is 204,800-dimensional, and also needs a non-linear kernel (Hellinger mapping) to achieve an accuracy that is still *significantly* lower than CNN. Clearly, CNNs are able to produce a very low-dimensional, highly discriminative, invariant and powerful representation for a given image.

Next, CNN descriptors obtained from rectified representations are also very powerful, and just as discriminative as SIFT (Table 6.4) or HOG (Table 6.4). The performance is understandably lower than a regular CNN representation since we have used a precise configuration of the detector, resulting in low recall as well as a sparser image representation. Moreover, an element-wise max operation on descriptors extracted from rectified patches performs better than a sum operation. This is also easily understood since a rectification always contains feature-less regions not originally present in a detection, consequently resulting in spurious features along the edges of the featured and featureless regions. Moreover, a max operation may also be thought of as selecting the largest responses to CNN features from any

[2]Comparing with previous works using CNN features for off-the-shelf description, note that the performance obtained here is slightly higher than that reported previously by [15] (FC-CNN in Table 2.1, 67.6% using the VGG-M pre-trained model), and slightly lower than by [107] (CNNaug-SVM in Table 2.1, 69% using the OverFeat model, but additional augmented training images).

Combined Rep.	**% Accuracy**
CNN + CNN_Rect(max) + SIFT	$75.96 \pm 0.18\%$
CNN + CNN_Rect(max) + SIFT_Rect	$75.54 \pm 0.17\%$
CNN + CNN_Rect(max) + SIFT + SIFT_Rect	$\mathbf{76.31 \pm 0.21}\%$

TABLE 6.8: MIT Indoor67 classification performance improvement with dense SIFT and off-the-shelf deep CNN feature description of affine-rectified texture.

overlapping, rectified patches, thereby retaining the more representative features.

It is seen that a combined regular and rectified representation can provide an improvement of up to almost 5%. This is a significant and impressive improvement, given that regular CNN would be expected to have already encoded a highly invariant representation (and given it has been trained on 1.2 million hand-labeled images of objects)! But the results here suggest an explicit planar rectification can still help push performance further. This also shows that the approach advocated in this thesis is not limited to hand-crafted features, but also extends to features extracted based on a powerful learning paradigm such as deep ConvNets.

Finally, Tables 6.8 and 6.9 present classification results based on various combinations of regular and rectified CNN, SIFT and HOG descriptors. The **best MIT Indoor67 classification accuracy achieved by this thesis is 76.90%**, which surpasses most current state-of-art approaches [Tables 2.1 and 2.2].

Combined Rep.	**% Accuracy**
CNN + CNN_Rect(max) + HOG	$76.19 \pm 0.23\%$
CNN + CNN_Rect(max) + HOG_Rect	$76.02 \pm 0.16\%$
CNN + CNN_Rect(max) + HOG + HOG_Rect	$\mathbf{76.90 \pm 0.47}\%$

TABLE 6.9: MIT Indoor67 classification performance improvement with dense HOG and off-the-shelf deep CNN feature description of affine-rectified texture.

6.2.6 Discussion

Table 6.10 presents per-class classification performance using regular and rectified SIFT and HOG features. Bold values indicate an increase in performance over the regular features, while a red value indicates a decrease. In general, categories that tend to exhibit homogeneous texture perform better upon rectification (SIFT_Rect + HOG_Rect) compared to before (SIFT + HOG), and the combination indicates complementary performance. The likely texture facilitating rectification, and consequently contributing toward performance improvement, stems from elaborate interiors and ceilings in "airport_inside", rows of seating in "auditorium" and "movietheater", patterned tiling in "bathroom", shelves and bookcases in "bookstore" and "library", lanes in "bowling" and "pool_inside", cribs in "nursery", etc.

The proposed approach does suffer from occasional failures. In this regard, examining some of the mis-classified images in Fig. 6.1 provides some insight into the confusions, which are more often than not quite plausible. The "auditorium" image indeed depicts non-conventional seating, more similar to lanes in "bowling_alley". Similarly, confusing a not-so-expansive "concert_hall" as "auditorium" is plausible, while the second confusion is likely due to the woodwork ceiling, more characteristic of outdoor structures such as a "greenhouse". Other confusions committed also seem plausible — the mis-classified "laundromat" images indeed lack the characteristic repeating patterns composed of laundry machines, while the first image is indeed more "kitchen"-like. Similarly, while the more typical "prison_cell" images with railings and bars were correctly classified, the ones depicting bunks and upholstery are mis-classified as, e.g., "living_room". Such features, not belonging to uniform patterns, are also not efficiently captured during dictionary learning, which is primarily based on features from homogeneous textured regions.

#	REP./ CATEGORY	SIFT+HOG	SIFT_Rect+ HOG_Rect	SIFT+HOG+ SIFT_Rect+ HOG_Rect+
01	airport_inside	0.40	**0.50**	**0.50**
02	art_studio	0.20	**0.35**	**0.30**
03	auditorium inside	0.72	**0.78**	0.72
04	bakery	0.26	**0.37**	**0.37**
05	bar	0.39	0.33	0.39
06	bathroom	0.56	**0.78**	**0.78**
07	bedroom	0.62	0.43	0.52
08	bookstore	0.50	**0.55**	0.50
09	bowling	0.95	**1.00**	0.95
10	buffet	0.75	0.75	0.75
11	casino	0.84	0.84	**0.89**
12	children_room	0.39	0.28	0.39
13	church_inside	0.74	**0.79**	0.68
14	classroom	0.67	**0.72**	**0.78**
15	cloister	0.95	**1**	0.95
16	closet	0.83	0.83	0.83
17	clothing_store	0.61	0.50	0.56
18	computer_room	0.72	**0.83**	**0.78**
19	concert_hall	0.80	0.75	0.70
20	corridor	0.57	**0.67**	**0.67**
21	deli	0.05	**0.11**	0.05
22	dental_office	0.67	0.62	0.62
23	dining_room	0.50	0.39	0.50
24	elevator	0.95	0.90	0.95
25	fastfood_restaurant	0.59	**0.71**	0.59
26	florist	0.95	0.79	0.89
27	gameroom	0.50	**0.65**	**0.65**
28	garage	0.72	0.67	**0.78**
29	greenhouse	0.85	0.85	0.85
30	grocery_store	0.57	**0.62**	**0.62**
31	gym	0.67	**0.89**	**0.83**
32	hair_salon	0.48	**0.62**	**0.57**
33	hospital_room	0.85	0.70	0.85
34	inside_bus	0.96	0.83	0.87

TABLE 6.10: Per-class classification performance for MIT Indoor67 with regular, rectified and combined gradient descriptors — 1/2.

#	REP./ CATEGORY	SIFT+HOG	SIFT_Rect+ HOG_Rect	SIFT+HOG+ SIFT_Rect+ HOG_Rect+
35	inside_subway	0.95	0.95	**1.00**
36	jewellery_shop	0.36	**0.50**	**0.55**
37	kindergarten	0.85	0.75	0.75
38	kitchen	0.62	**0.71**	**0.67**
39	laboratory_wet	0.55	0.36	0.50
40	laundromat	0.82	0.73	0.82
41	library	0.50	0.50	**0.60**
42	living_room	0.30	**0.40**	0.30
43	lobby	0.25	**0.40**	**0.35**
44	locker_room	0.57	0.38	0.48
45	mall	0.65	0.65	0.65
46	meeting_room	0.68	0.68	**0.73**
47	movie_theater	0.70	**0.80**	0.70
48	museum	0.43	0.43	0.43
49	nursery	0.75	0.75	**0.80**
50	office	0.05	**0.10**	**0.19**
51	operating_room	0.37	**0.47**	**0.53**
52	pantry	0.80	**0.85**	**0.85**
53	pool_inside	0.65	**0.70**	**0.70**
54	prison_cell	0.70	0.65	**0.75**
55	restaurant	0.55	0.40	0.50
56	restaurant_kitchen	0.61	0.57	0.61
57	shoeshop	0.58	0.58	0.58
58	staircase	0.80	0.75	0.80
59	studio_music	0.84	**0.89**	**0.89**
60	subway	0.57	0.52	**0.62**
61	toystore	0.27	0.27	**0.32**
62	train_station	0.75	**0.80**	**0.80**
63	tv_studio	0.78	0.72	0.72
64	video_store	0.59	0.50	0.50
65	waiting_room	0.38	**0.43**	**0.52**
66	warehouse	0.57	0.57	**0.62**
67	wine_cellar	0.81	0.81	0.81
	MEAN	0.62	**0.63**	**0.65**

TABLE 6.10: Per-class classification performance for MIT Indoor67 with regular, rectified and combined gradient descriptors — 2/2.

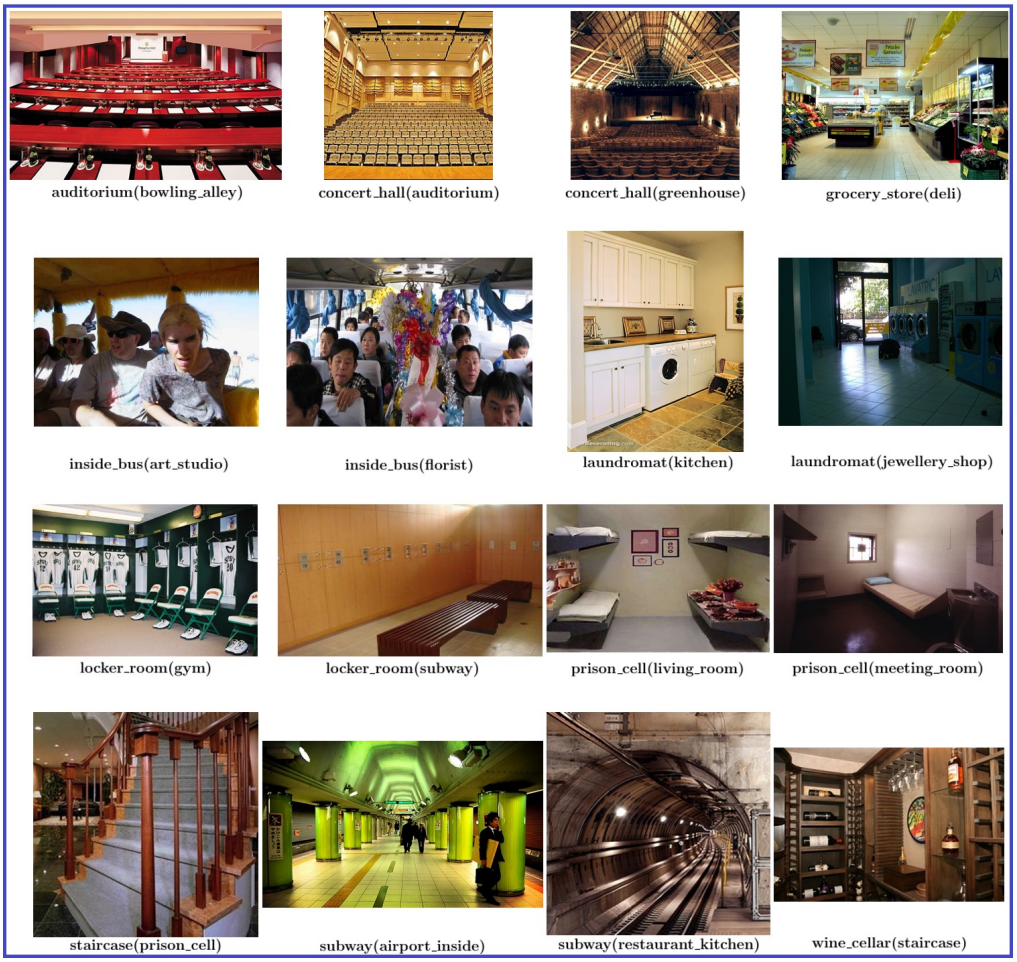


FIGURE 6.1: Sample MIT Indoor67 test images that were mis-classified when using a representation based on affine-rectified homogeneous texture, but correctly classified when using a regular representation, in format [true_category(assigned_category)].

Fig. 6.2 on the other hand presents example images that were originally mis-classified using a regular representation, but a texture-rectified representation helped facilitate a correct classification. Typical "airport_inside" images with textured walls and ceilings, and lacking explicit rail-tracks are correctly classified. A homogeneous texture based representation focuses more on the rows of desks in a "classroom". On the other hand, a regular representation would also consider the blackboard as important, and

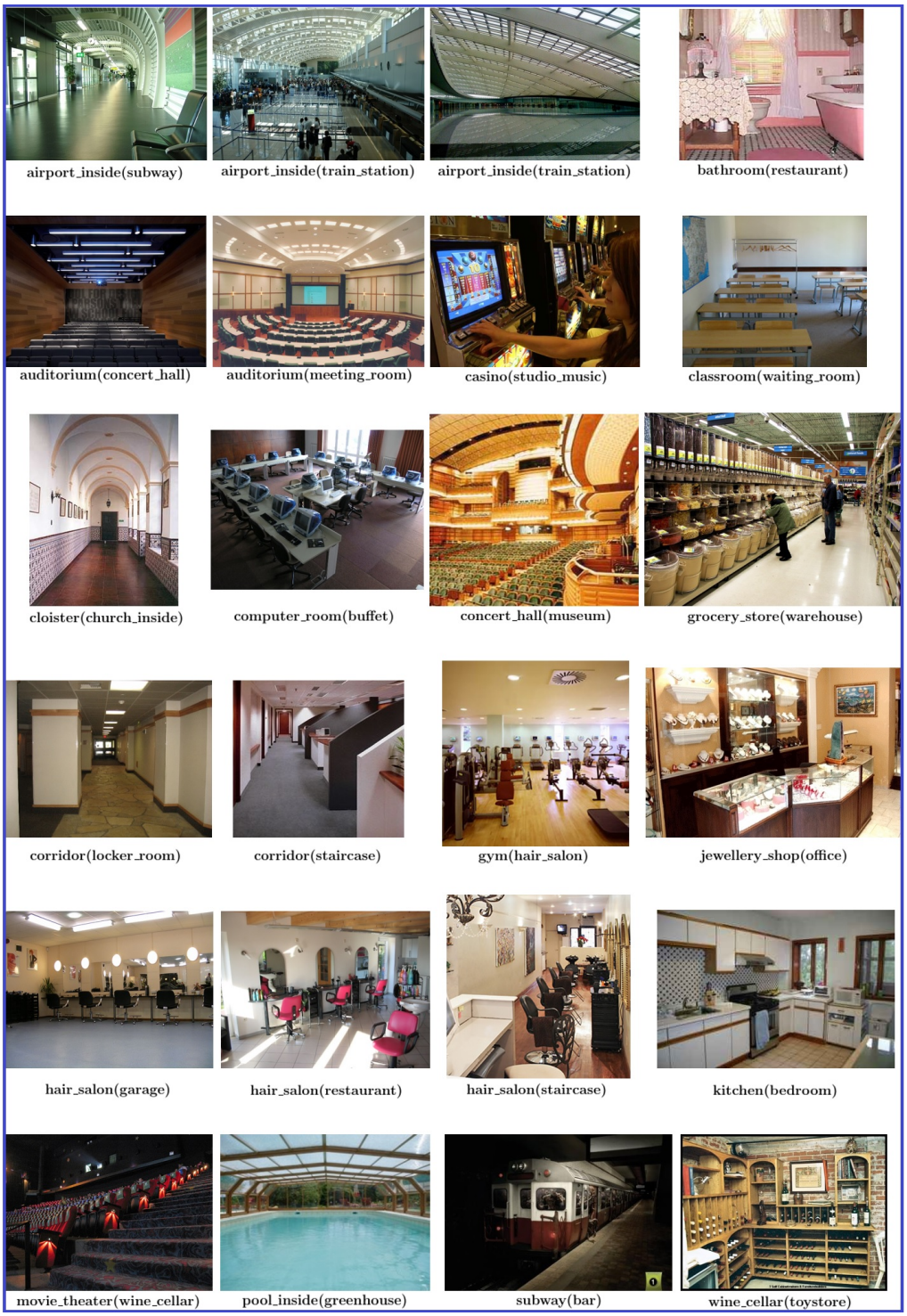


FIGURE 6.2: Sample MIT Indoor67 test images that were mis-classified when using a regular representation, but correctly classified when using a representation based on affine-rectified homogeneous texture, in format [true_category(assigned_category)].

in the absence of it, assigns this image to "waiting_room" (which also contains seating, but *not* rows of desks, hence the texture based representation seems undeterred). Similar observations can be made for the rest of these examples. A notable property among most of them is large perspective distortion, as well as uniform texture.

6.3 Experiments on Places2 [150] Subset

While the primary aim of this thesis was to facilitate and investigate the role of texture in indoor scene recognition, this section is dedicated to evaluating the proposed approach on a broader range of scenes. Specifically, a subset of the Places2 scene dataset ([150], Sec. 2.10) is considered, consisting of 31 scene categories — 1 natural, 5 indoor and the remaining 25 being man-made outdoor environments, all of which tend to exhibit regular, repeating structure (see Table 6.15). Moreover, for each category, the first 150 training images are used, while testing is done on the 50 validation images. This makes for a subset of 6200 images (similar in size to the MIT Indoor67, which contains 6700 images).

Fig. 6.3 shows qualitative results of detection on some sample images from various Places2 categories. Tables 6.11, 6.12, 6.13, 6.14 and 6.15 present classification performance with CENTRIST, LBP, SIFT, HOG and CNN features, respectively. As with the MIT Indoor67 dataset, we find that rectified representations are not only discriminative but also complement regular representations.

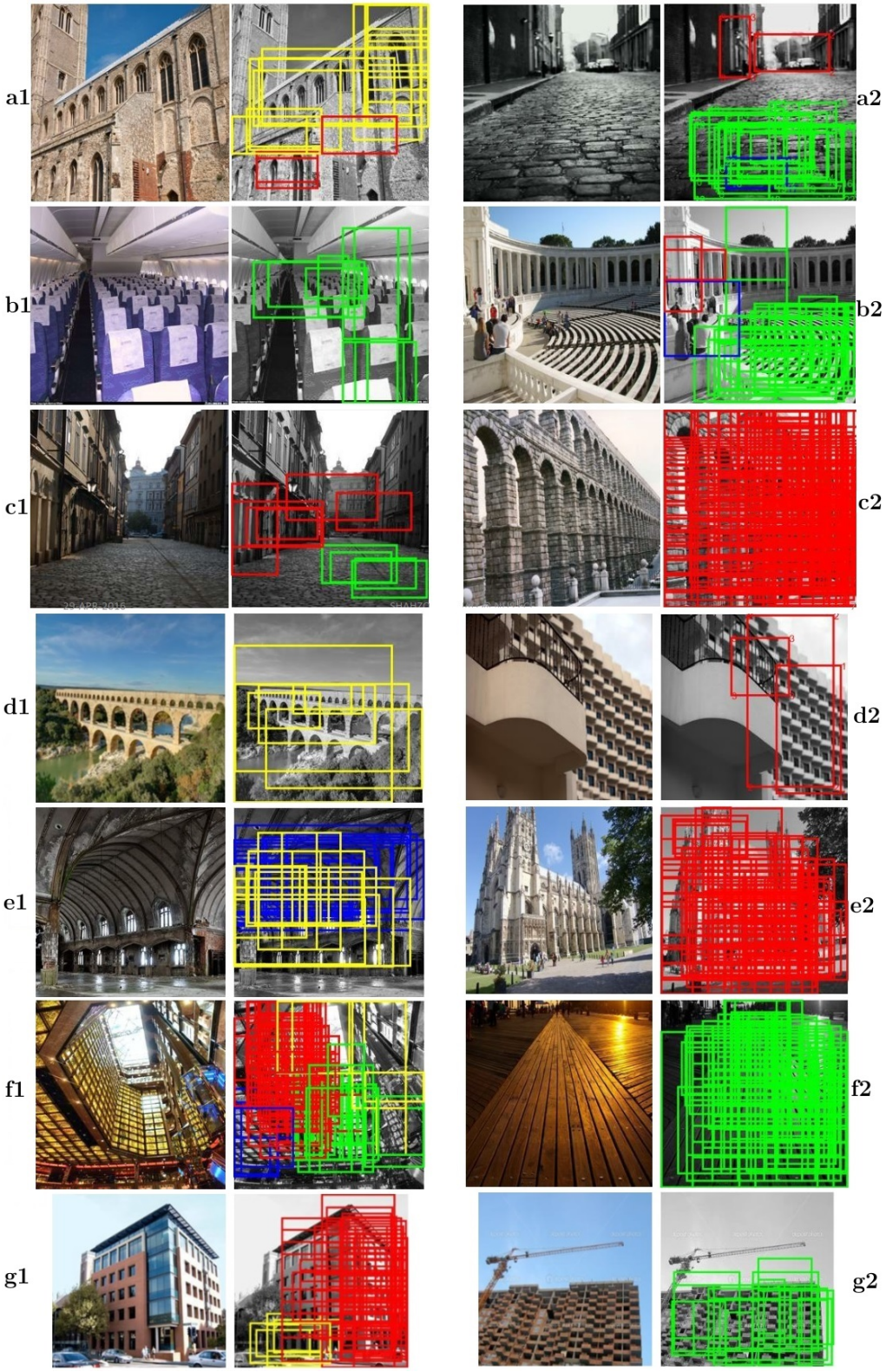


FIGURE 6.3: Homogeneous texture detection and its geometric class assignment on images from various Places2 [150] scene dataset categories — 1/2: (a1) abbey, (a2) alley, (b1) airplane_cabin, (b2) amphitheater, (c1) alley, (c2) aqueduct, (d1) aqueduct, (d2) balcony, (e1) arch, (e2) basilica, (f1) atrium, (f2) boardwalk, (g1) campus, (g2) construction_site.

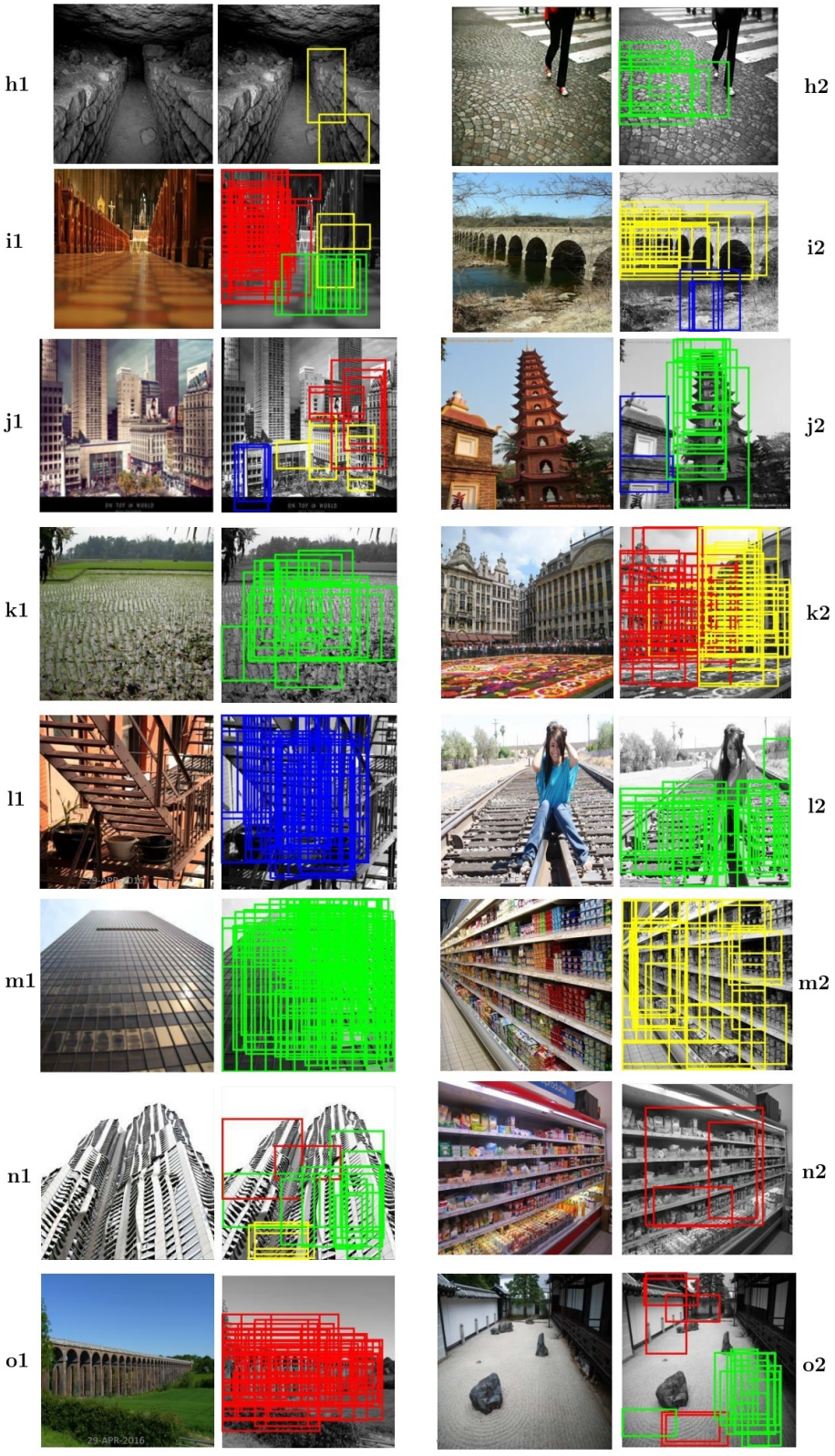


FIGURE 6.3: Homogeneous texture detection and its geometric class assignment on images from various Places2 [150] scene dataset categories — 2/2: (h1) catacomb, (h2) crosswalk, (i1) cathedral_indoor, (i2) dam, (j1) downtown, (j2) pagoda, (k1) field_cultivated, (k2) plaza, (l1) fire_escape, (l2) railroad_track, (m1) skyscraper, (m2) supermarket, (n1) skyscraper, (n2) supermarket _site, (o1) viaduct, (o2) zen_garden.

Rep.	**% Accuracy**
CEN	$46.69 \pm 0.62\%$
CEN_Rect	$\mathbf{47.38 \pm 0.20}\%$
CEN + CEN_Rect	$\mathbf{48.77 \pm 0.07}\%$

TABLE 6.11: Places2 subset classification performance improvement with dense feature description of affine-rectified texture — CENTRIST.

Rep.	**% Accuracy**
LBP(16,2)_u2	$41.55 \pm 0.34\%$
LBP(16,2)_u2_Rect	$\mathbf{45.38 \pm 0.62}\%$
LBP(16,2)_u2 + LBP(16,2)_u2_Rect	$\mathbf{46.01 \pm 0.23}\%$

TABLE 6.12: Places2 subset classification performance improvement with dense feature description of affine-rectified texture — Local Binary Patterns LBP^{u2}.

Rep.	**% Accuracy**
SIFT	$54.93 \pm 0.33\%$
SIFT_Rect	$54.84 \pm 0.42\%$
SIFT + SIFT_Rect	$\mathbf{56.13 \pm 0.42}\%$

TABLE 6.13: Places2 subset classification performance improvement with dense feature description of affine-rectified texture — SIFT.

Rep.	**% Accuracy**
HOG	$53.91 \pm 0.53\%$
HOG_Rect	$\mathbf{55.27 \pm 0.65}\%$
HOG + HOG_Rect	$\mathbf{56.17 \pm 0.61}\%$

TABLE 6.14: Places2 subset classification performance improvement with dense feature description of affine-rectified texture — HOG.

Rep.	**% Accuracy**
CNN	63.03%
CNN_Rect(sum)	51.48%
CNN_Rect(max)	51.68%
CNN + CNN_Rect(sum)	**63.94**%
CNN + CNN_Rect(max)	**64.58**%

TABLE 6.15: Places2 subset classification performance improvement with off-the-shelf deep CNN feature description of affine-rectified texture.

6.3.1 Discussion

The performance improvement for CNN descriptors with a combined representation is not as pronounced as for the MIT Indoor67, even though all these categories exhibit homogeneous texture. This section attempts to explain this behaviour. Table 6.15 shows the class-wise performance for the regular, rectified and combined representation. It is seen that most of the categories benefit when the two representations are used in conjunction, and the few drops in performance are also minor. However, the rectified representation on its own mostly fails to perform good classifications, and is essentially the reason why the overall performance improvement is not very impressive. To understand why, Fig. 6.4 analyzes sample images from five categories with the most drastic decrease in performance by a rectified representation over the regular one (namely: "alley", "atrium", "dam", "field_cultivated" and "railroad_track").

Perhaps unsurprisingly, nearly all the confusions committed are highly plausible. Some categories in this dataset have only subtle differences and it is easy to mistake one for the other, especially in the absence of some sort of high-level contextual reasoning. Examples include "dam", "aqueduct" and "viaduct", as well as "field_cultivated", "formal_garden" and "bamboo_forest". Some of the confusions arise out of the representation's focus

on characteristic homogeneous texture and this can explain, for example, the "alley" images mistaken as "arcade" or "railroad˙track", the "atrium" images mistaken as "balcony_exterior" or "cathedaral_indoor", or the wood-constructed "dam" mistaken as a "boardwalk". For some examples, it is arguably difficult for even humans to assign a unique category — for example, the 3rd "alley" image, the 5th "dam" image, or the 3rd "field_cultivated" image. Nevertheless, the fact that a combined regular and rectified representation provides some overall performance improvement suggests that multiple scene cues can indeed help improve classification, and there is a need to research such cues as well as more principled approaches to combining and exploiting them.

Fig. 6.5 presents additional insightful examples depicting a mix of success and failure cases for both a regular and rectified CNN representation. We again observe that some of these cases can indeed be assigned multiple scene categories (e.g., "downtown" and "skyscraper"), while for some a correct categorisation is difficult to achieve without some high-level visual reasoning. Indeed, considering the large scale nature of this dataset [150] it has been suggested that an algorithm be allowed to produced up to 5 possible category labels for any given test image at the ILSVRC challenge [113],

#	REP./ CATEGORY	CNN	CNN_Rect(max)	CNN+ CNN_Rect(max)
01	abbey	0.60	0.26	0.60
02	airplane_cabin	1	0.94	1
03	alley	0.74	0.46	**0.78**
04	amphitheater	0.7	0.62	**0.76**
05	aqueduct	0.58	0.58	0.58
06	arcade	0.44	0.38	**0.50**
07	arch	0.20	0.06	**0.24**
08	atrium	0.72	0.34	0.70
09	balcony_exterior	0.44	0.34	0.42
10	bamboo_forest	0.84	**0.90**	**0.92**
11	basilica	0.32	0.32	**0.38**
12	boardwalk	0.58	0.32	**0.60**
13	campus	0.38	0.24	0.38
14	catacomb	0.82	0.80	0.82
15	cathedral_indoor	1	0.90	1
16	construction_site	0.6	0.46	0.58
17	courthouse	0.58	0.44	**0.62**
18	crosswalk	0.88	0.72	0.84
19	dam	0.72	0.52	**0.74**
20	downtown	0.30	0.24	0.30
21	field_cultivated	0.90	0.66	0.90
22	fire_escape	0.70	0.70	**0.74**
23	formal_garden	0.76	**0.84**	**0.78**
24	pagoda	0.70	0.52	0.70
25	plaza	0.14	0.08	**0.16**
26	railroad_track	0.78	0.50	0.76
27	shopfront	0.96	0.88	**0.98**
28	skyscraper	0.30	**0.34**	**0.38**
29	supermarket	1	0.98	1
30	viaduct	0.24	0.14	0.22
31	zen_garden	0.62	0.54	**0.64**
	MEAN	0.63	0.52	**0.65**

TABLE 6.15: Per-class classification performance for Places2 subset with regular, rectified and combined ConvNet descriptors.

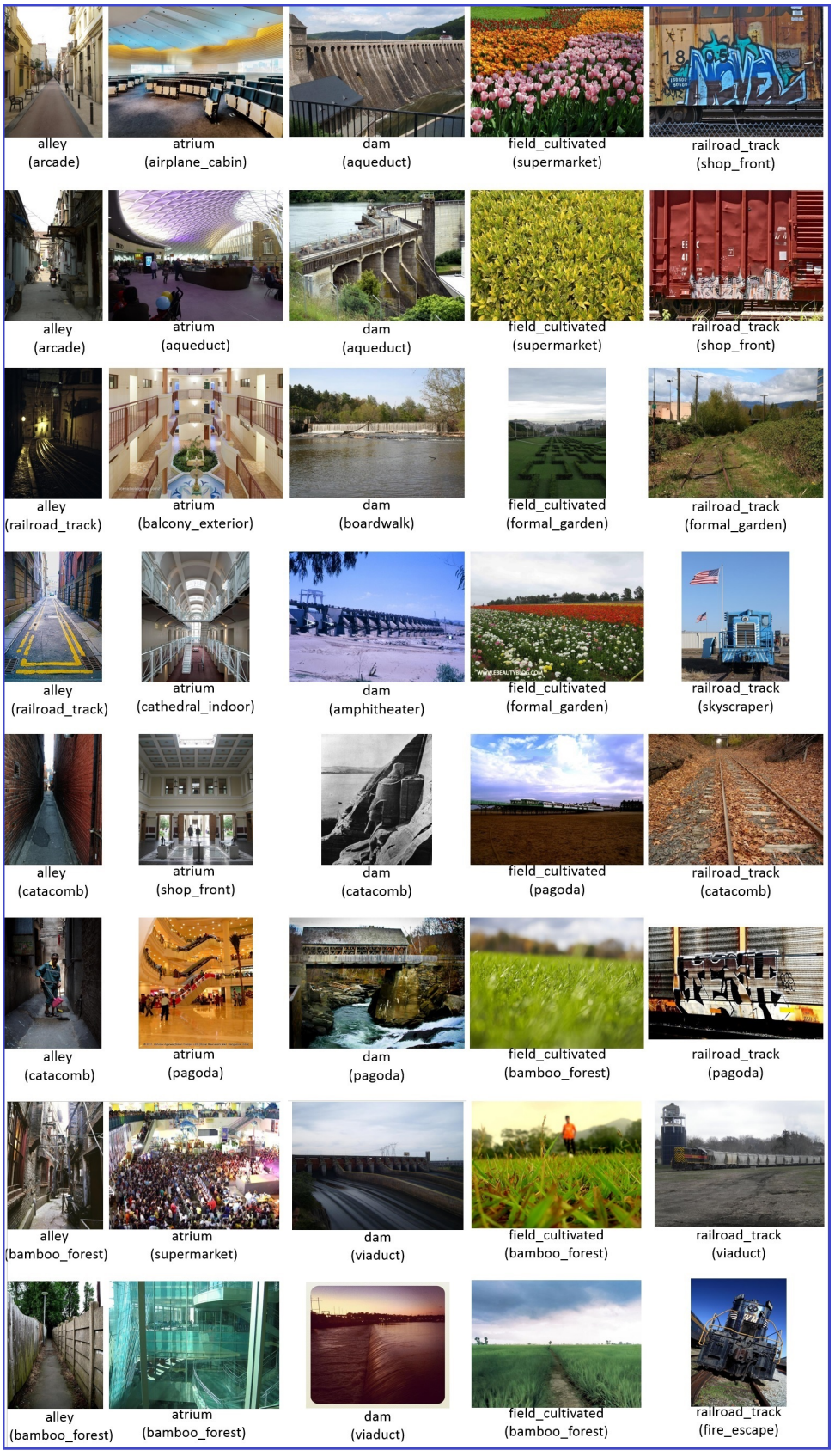


FIGURE 6.4: Sample Places2 validation images that were mis-classified when using a ConvNet representation based on affine-rectified homogeneous texture, but correctly classified when using a regular ConvNet representation, in format [true_category(assigned_category)].

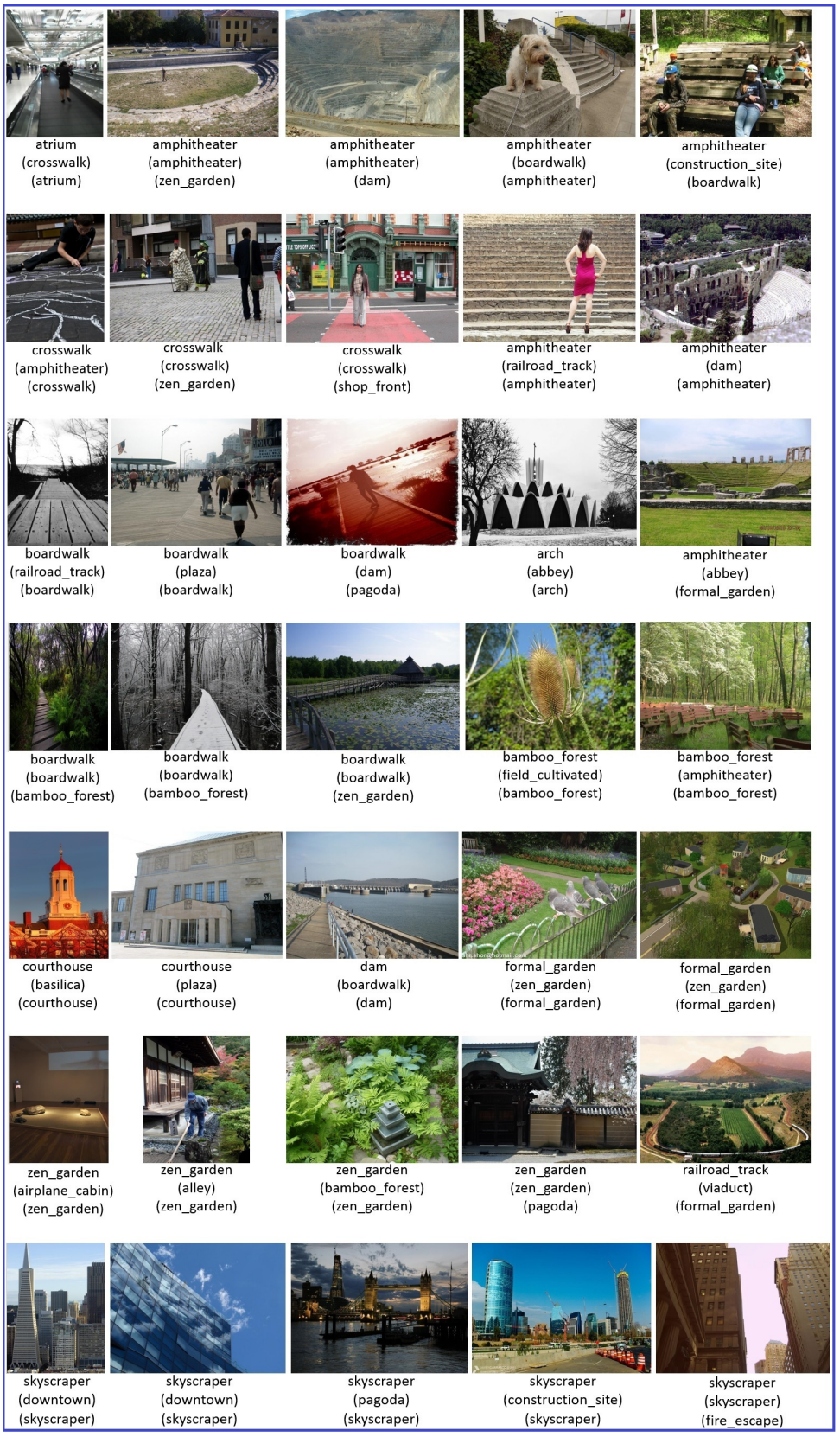


FIGURE 6.5: Sample Places2 validation images with assigned category using a regular ConvNet representation, or that based on affine-rectified homogeneous texture indicated in format [true_category(assigned_category <regular>)(assigned_category <rectified>)].

Chapter 7

Conclusions & Future Work

7.1 Conclusions

This thesis has advanced a novel paradigm involving the use of homogeneous texture — widely manifested in indoor scenes — for an improved scene understanding and classification (see Sec. 4.1, 5.1). It thus deviates from the established practice of employing machine learning in order to estimate scene layouts (Sec. 3.2) or to extract features for recognition (Sec. 3.1).

A mathematical model has been developed in Chapter 4 that allows the recovery of plane projective parameters from imaged texture, facilitating an affine rectification. Robust methods to measure the dominant instantaneous frequency in imaged texture are developed (Sec. 4.4), and robust recovery of projective parameters demonstrated (Sec. 4.5). The resulting frequency based approach is shown to outperform existing representative methods on the task of rectification of real-world texture (Sec. 4.7).

The texture projection model is then applied to detecting homogeneous textured regions in real-world, cluttered indoor scenes in Sec. 5.2. This facilitates the estimation of the geometric layout in multi-planar textured indoor scenes (Sec. 5.3). In doing so, the approach sidesteps the error-prone, ill-posed computation of vanishing points in order to establish room orientation, and does not need to rely upon the simplistic Manhattan or box layout assumption, or to employ machine learning to localize room faces in space and scale.

Affine rectification of detected homogeneous texture is found to yield low-level features that are not only class-discriminative, but also complementary to regular, non-rectified features, thereby facilitating indoor scene recognition (Chapter 6). The results are consistent across a number of hand-crafted descriptors, both thresholding (CENTRIST, LBP) and gradient based (SIFT, HOG), as well as pre-learned deep ConvNet features. Classification performance based on a combined feature representation is seen to favorably compare with contemporary approaches on the MIT Indoor67 benchmark, while one of the presented configurations outperforms most current state-of-the-art work. The proposed approach is additionally evaluated on a set of 6200 (mostly outdoor) images, being a subset of the Places2 large scale scene dataset.

In summary, the thesis attempts to draw attention of the community toward the role of a particular, abundantly occurring class of texture — that which satisfies the homogeneity assumption — in describing indoor scenes, and consequently facilitating their semantic recognition. It is an effort toward ironing out some of the technical challenges that would otherwise prevent a successful use of such texture in performing scene classification in real-world images, thereby paving the way for further research in this promising direction.

7.2 Future Work

A limitation of the proposed texture projection model is that the error measures defined in Eqns. 4.9 and 4.11 are not affine invariant. A heuristic normalization strategy has, however, been presented in Sec. 4.6, and is seen to perform very well for texture rectification (where it serves to select the best multi-scale representation), texture detection, and consequently scene layout estimation and classification. However, the current occasional failures can be significantly mitigated, and classification performance pushed further, should an invariant error measure be discovered.

Like generic low-level (blobs and edges) or mid-level features (distinctive scene parts), homogeneous texture is sparsely manifested in scenes. As such, the experiments presented in this thesis have made use of an existing classifier score fusion scheme to complement features from homogeneous texture with regular densely extracted features. Higher performance may be achieved, however, and more insight attained as to what scene categories can be well described by homogeneous texture, by devising schemes that can more effectively leverage the complementarity of regular and texture based features. The thesis has also attempted to bring to light the complementary nature of various gradient and threshold based hand-crafted descriptors, as well as pre-trained deep ConvNet features and more work to further explore this synergy might prove fruitful.

The warping process in rectification gives rise to artifacts in regions which are magnified (resulting in oversampling) or minified (leading to undersampling, and hence aliasing) with respect to the original image [53] (see, e.g., Fig. 4.15(a,n)). Such artifacts in the process of rectification likely lead to lower performance than can potentially be attained, hence must be addressed.

Since the obtained rectifications still manifest an unknown affine transform, rotating the descriptors such as by SIFT [84], or affine adaptation such as by [88, 64] may be explored to further improve performance. Alternatively, it might also be worthwhile to investigate the use of affine-invariant texture signatures (e.g. [147]), computed directly from imaged texture. Other potential avenues to explore the role of texture in recognition include fractal and lacunarity (see e.g., [105]) analysis on detected texture, or to employ deep ConvNet learning for texture detection or recognition [15].

Bibliography

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, 54(11):4311 – 4322, 2006. 18

[2] T. Ahonen, J. Matas, C. He, and M. Pietikinen. Rotation invariant image description with Local Binary Pattern Histogram Fourier Features. In *Proc. 16th Scandinavian Conference on Image Analysis*, 2009. 140

[3] D. Aiger, D. Cohen-Or, and N. J. Mitra. Repetition maximization based texture rectification. *Computer Graphics Forum (EUROGRAPHICS)*, 31(2pt2):439–448, 2012. 62, 63, 92, 94, 95, 96

[4] R. Arandjelovi and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2911 – 2918, 2012. 143

[5] O. Boiman, E. Shechtman, and M. Irani. In defense of Nearest-Neighbor based image classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1 – 8, 2008. 24

[6] A. Bosch, A. Zisserman, and X. Munoz. Scene classification via pLSA. In *Proc. European Conf. on Computer Vision*, pages 517–530, 2006. 12, 24, 26

[7] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2559 – 2566, 2010. 21, 22, 23

[8] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001. 77, 79

[9] F. Cakir, U. Gudukbay, and O. Ulusoy. Nearest-neighbor based metric functions for indoor scene recognition. *Computer Vision and Image Understanding*, 115(11):1483–1492, 2011. 24

[10] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. 138

[11] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *Proc. British Machine Vision Conference*, pages 76.1–76.12, 2011. 17, 24, 44, 137

[12] M. J. Choi, J. Lim, A. Torralba, and A. Willsky. Exploiting hierarchical context on a large database of object categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 129 – 136, 2010. 44

[13] M. J. Choi, J. Lim, A. Torralba, and A. Willsky. Exploiting hierarchical context on a large database of object categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 129 – 136, 2010. 46

[14] O. Chum and J. Matas. Planar affine rectification from change of scale. In *Proc. Asian Conf. on Computer Vision*, pages 347–360, 2010. 62, 63, 64

[15] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recongition and segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3828–3836, 2015. 35, 39, 41, 146, 165

[16] T. Collins, J. Durou, P. Gurdjos, and A. Bartoli. Single-view perspective shape-from-texture with focal length estimation: A piecewise affine approach. In *Proc. 3D Data Processing, Visualization and Transmission (3DPVT)*, 2010. 61, 64

[17] J. M. Coughlan and A. L. Yuille. Manhattan world: compass direction from a single image by Bayesian inference. In *Proc. IEEE International Conf. on Computer Vision*, pages 941 – 947, 1999. 52

[18] A. Criminsi and A. Zisserman. Shape from texture: homogeneity revisited. In *Proc. British Machine Vision Conference*, page 8291, 2000. 61, 63

[19] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE International Conf. on Computer Vision*, pages 886 – 893, 2005. 10, 13

[20] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1160–1169, 1985. 70

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 248 – 255, 2009. `http://www.image-net.org/`. 28, 36, 44

[22] C. Desai, D. Ramanan, and C. C. Fowlkes. Discriminative models for multi-class object layout. In *Proc. IEEE International Conf. on Computer Vision*, pages 229 – 236, 2009. 44

[23] C. Doersch, A. Gupta, and A. A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *Proc. Neural Information Processing Systems*, pages 494–502, 2013. 32, 41, 42, 45, 46, 100

[24] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes Paris look like Paris? *ACM Trans. on Graphics (SIGGRAPH 2012)*, 31(4):101:1–101:9, 2012. 30, 32, 45

[25] J. Donahue*, Y. Jia*, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *Proc. International Conf. on Machine Learning*, 2014. (* = equal contribution). 38, 41

[26] G. Dorko and C. Schmid. Object class recognition using discriminative local features. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2005. 12, 17

[27] K. A. Ehinger, J. Xiao, A. Torralba, and A. Oliva. Estimating scene typicality from human ratings and image features. In *Proc. 33rd Annual Meeting of the Cognitive Science Society*, 2011. 9

[28] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2014. 129, 130

[29] E. Farahzadeh, T.-J. Cham, and W. Li. Incorporating local and global information using a novel distance function for scene recognition. In *IEEE Workshop on Robot Vision (WORV)*, pages 132 – 137, 2013. 10

[30] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE Conf. on Computer Vision and Pattern Recognition Workshop*, page 178, 2004. 12, 15

[31] L. Fei-Fei, A. Iyer, C. Koch, and P. Perona. What do we perceive in a glance of a real-world scene? *Journal of Vision*, 7(1):1–29, 2007. 9

[32] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 524–531, 2005. 12, 15, 24, 26, 27, 39

[33] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 10, 13, 29, 31, 32, 105

[34] J. Feng, B. Ni, Q. Tian, and S. Yan. Geometric Lp-norm feature pooling for image classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2609 – 2704, 2011. 23, 40

[35] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 264–271, 2003. 12

[36] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 85, 107

[37] A. Fitzgibbon. "Andrew Zisserman, BMVA Distinguished Fellow 2008". `http://www.bmva.org/2008_zisserman`, 2008. 1

[38] M. Fornoni and B. Caputo. Indoor scene recognition using task and saliency-driven feature pooling. In *Proc. British Machine Vision Conference*, pages 98.1–98.12, 2012. 24

[39] S. Gao, I. Tsang, L.-T. Chia, and P. Zhao. Local features are not lonely Laplacian sparse coding for image classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3555–3561, 2010. 16

[40] J. Garding. Shape from texture and contour by weak isotropy. *Artificial Intelligence*, 64(2):243–297, 1993. 63

[41] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *Proc. European Conf. on Computer Vision*, pages 392–407, 2014. 39, 41

[42] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. url=http://www.deeplearningbook.org, 2016. Book in preparation for MIT Press. 36

[43] K. Grauman and T. Darrell. The Pyramid Match Kernel: Discriminative classification with sets of image features. In *Proc. IEEE International Conf. on Computer Vision*, pages 1458–1465, 2005. 15

[44] M. R. Greene and A. Oliva. Recognition of natural scenes from global properties: seeing the forest without representing the trees. *International Journal of Computer Vision*, 58(2):137–176, 2009. 9

[45] P. Gupta, S. Arrabolu, M. Brown, and S. Savarese. Video scene categorization by 3d hierarchical histogram matching. In *Proc. IEEE International Conf. on Computer Vision*, pages 1655–1662, 2009. 15

[46] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *Proc. European Conf. on Computer Vision*, pages 459–472, 2012. 32

[47] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 51, 66, 118

[48] T. Hassner, V. Mayzels, and L. Zelnik-Manor. On SIFTs and their scales. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1522 – 1528, 2012. 13

[49] J. P. Havlicek, A. C. Bovik, and P. Maragos. Modulation models for image processing and wavelet-based image demodulation. In *Proc.*

Asilomar Conf. on Signals, Systems and Computers, pages 805 – 810, 1992. 61, 72

[50] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 35, 38

[51] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *Proc. IEEE International Conf. on Computer Vision*, pages 1849 – 1856, 2009. xii, xiv, 51, 52, 53, 54, 55, 61, 116, 119, 121, 122, 123, 124

[52] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007. xii, xvii, 48, 49, 51, 52, 54, 124

[53] D. H. House. Avoiding artifacts in warped images. http://people.cs.clemson.edu/~dhouse/courses/405/notes/antialiasing.pdf, Retrieved Jan 2016. 164

[54] Y. Huang, Z. Wu, L. Wang, and T. Tan. Feature coding in image classification: A comprehensive study. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(3):493 – 506, 2014. 17, 44

[55] P. Isola, J. Xiao, A. Torralba, and A. Oliva. What makes an image memorable? In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 145–152, 2011. 13

[56] H. Izadinia, F. Sadeghi, and A. Farhadi. Incorporating scene context and object layout into appearance modeling. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 232 – 239, 2014. 44

[57] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20, 2015. 35

[58] S. Jalali, J. Lim, S. Ong, and J. Tham. Realistic modeling of simple and complex cell tuning in the HMAX Model, and implications for

invariant object recognition in cortex. *Neural Information Processing. Models and Applications (LNCS)*, 6444:541–548, 2010. 25

[59] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *Proc. IEEE International Conf. on Computer Vision*, pages 2146–2153, 2009. 37

[60] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3370 – 3377, 2012. 23

[61] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. http://caffe.berkeleyvision.org/. 38

[62] A. Jiang, C. Wang, B. Xiao, and R. Dai. A new biologically inspired feature for scene image classification. In *Proc. International Conf. on Pattern Recognition*, pages 758 – 761, 2010. 25, 40

[63] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 923 – 930, 2013. 18, 32, 41, 42, 45, 46, 100, 106, 137, 142, 143

[64] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *Proc. European Conf. on Computer Vision*, pages 228–241, 2004. 12, 104, 165

[65] Y. Karklin and M. S. Lewicki. Emergence of complex cell properties by learning to generalize in natural scenes. *Nature*, 457:83–86, 2008. 19, 26, 44

[66] V. Kolmogorov and C. Rother. Minimizing non-submodular functions with graph cuts - a review. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(7):1274–1279, 2007. 79

[67] J. Kosecka and W. Zhang. Extraction, matching and pose recovery based on dominant rectangular structures. In *First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis, 2003*, pages 83 – 91, 2003. 61

[68] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Neural Information Processing Systems*, pages 1097–1105, 2012. 35, 37, 38

[69] J. Krumm and S. Shafer. Shape from periodic texture using the spectrogram. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 284 – 289, 1992. 63, 64

[70] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006. 15, 20, 23, 24, 27, 40, 50, 137

[71] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. 34, 36

[72] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278 – 2324, 1998. 34, 36

[73] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 97 – 104, 2004. 37

[74] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *Proc. IEEE International Symposium on Circuits and Systems*, pages 253–256, 2010. 35

[75] H. Lee, A. Battle, R. Raina, , and A. Y. Ng. Efficient sparse coding algorithms. In *Proc. Neural Information Processing Systems*, pages 801 – 808, 2006. 18

[76] C. Li, D. Parikh, and T. Chen. Automatic discovery of groups of objects for scene understanding. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2735 – 2742, 2012. 47

[77] L.-J. Li, , R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2036 – 2043, 2009. 26

[78] L.-J. Li*, H. Su*, E. P. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *Proc. Neural Information Processing Systems*, pages 1378–1386, 2010. 28, 45

[79] X.-C. Lian, Z. Li, B.-L. Lu, and L. Zhang. Max-margin dictionary learning for multiclass image categorization. In *Proc. European Conf. on Computer Vision*, pages 157–170, 2010. 22

[80] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 482 – 488, 1998. 60, 62

[81] D. Lin, C. Lu, R. Liao, and J. Jia. Learning important spatial pooling regions for scene classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3726 – 3733, 2014. 30, 42

[82] X. Liu, O. Veksler, and J. Samarabandu. Order-preserving moves for graph-cut-based optimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(7):1182–1196, 2010. 124

[83] L. C. Loschky and A. M. Larson. Localized information is necessary for scene categorization, including the natural/man-made distinction. *Journal of Vision*, 8(1):1–9, 2008. 10

[84] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 10, 11, 12, 104, 165

[85] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1 – 8, 2008. 20

[86] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Proc. Neural Information Processing Systems*, pages 1033–1040, 2008. 20

[87] R. Margolin, L. Zelnik-Manor, and A. Tal. Otc: A novel local descriptor for scene classification. In *Proc. European Conf. on Computer Vision*, pages 377–391, 2014. 33

[88] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from Maximally Stable Extremal Regions. In *Proc. British Machine Vision Conference*, pages 36.1–36.10, 2002. 12, 104, 165

[89] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. 14

[90] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine

region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005. 14

[91] J. Mutch and D. G. Lowe. Object class recognition and localization using sparse features with limited receptive fields. *International Journal of Computer Vision*, 80(1):45–57, 2008. 22, 25, 71

[92] M. A. Nielsen. Neural networks and deep learning. http://neuralnetworksanddeeplearning.com, 2015. 35, 36

[93] T. Ojala, M. Pietikinen, and T. Menp. Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. 33, 139

[94] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. 8, 14, 15, 27, 34, 39

[95] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *Proc. IEEE International Conf. on Computer Vision*, pages 1307 – 1314, 2011. 10, 29, 30, 32, 44, 46, 100, 138

[96] S. Parizi, J. Oberlin, and P. Felzenszwalb. Reconfigurable models for scene recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2775–2782, 2012. 15, 24, 26

[97] G. Patterson, T.-Y. Lin, and J. Hays. Using humans to build mid-level features. In *IEEE Conf. on Computer Vision and Pattern Recognition Workshop*, 2013. 44

[98] G. Patterson, C. Xu, H. Su, and J. Hays. The SUN attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1):59–81, 2014. 34, 44

[99] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Occlusion patterns for object class detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3286 – 3293, 2013. 46

[100] F. Perronnin. Universal and adapted vocabularies for generic visual categorization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(7):1243–1256, 2008. 17

[101] F. Perronnin, J. Sanchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *Proc. European Conf. on Computer Vision*, pages 143–156, 2010. 17

[102] N. Petkov and P. Kruizinga. Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells. *Biological Cybernetics*, 76(2):83–96, 1997. 71, 73

[103] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 16

[104] J. Pritts, O. Chum, and J. Matas. Detection, rectification and segmentation of coplanar repeated patterns. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2973 – 2980, 2014. 62, 63, 64

[105] Y. Quan, Y. Xu, Y. Sun, and Y. Luo. Lacunarity analysis on image patterns for texture classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 160 – 167, 2014. 165

[106] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 413 – 420, 2009. ix, 9, 10, 27, 28, 40, 135, 138

[107] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition Workshop*, pages 512–519, 2014. 35, 38, 41, 146

[108] E. Ribeiro and E. R. Hancock. Estimating the 3d orientation of texture planes using local spectral analysis. *Image and Vision Computing*, 18(8):619–631, 2000. 62

[109] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025, 1999. 25

[110] R. Rigamonti, M. Brown, and V. Lepetit. Are sparse representations really relevant for image classification? In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1545 – 1552, 2011. 21

[111] R. Rosenholtz and J. Malik. Surface orientation from texture: isotropy or homogeneity (or both)? *Vision Resarch*, 37(16):2283–2293, 1997. 61, 63

[112] C. Rother. A new approach for vanishing point detection in architectural environments. In *Proc. British Machine Vision Conference*, pages 382–391, 2000. 51, 53

[113] O. Russakovsky*, J. Deng*, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. (* = equal contribution). xvii, 36, 37, 40, 41, 42, 158

[114] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *Proc. International Conference on*

Learning Representations, 2014. http://cilvr.nyu.edu/doku.php?id=software:overfeat:start. 38

[115] T. Serre and M. Riesenhuber. Realistic modeling of simple and complex cell tuning in the HMAX Model, and implications for invariant object recognition in cortex. *MIT CSAIL Tech. Report*, 2004. 25

[116] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 994 – 1000, 2005. 22, 25, 71

[117] D. Shaw and N. Barnes. Perspective rectangle detection. In *European Conference on Computer Vision Workshop on Applications of Computer Vision*, 2006. 61

[118] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. International Conference on Learning Representations*, 2015. 35, 37, 145

[119] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *Proc. European Conf. on Computer Vision*, pages 73–86, 2012. 30, 31, 32, 45, 46, 100, 105, 106

[120] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. IEEE International Conf. on Computer Vision*, pages 1470 – 1477, 2003. 12, 14

[121] X. Y. Stella, H. Zhang, and J. Malik. Inferring spatial layout from a single image via depth-ordered grouping. In *IEEE Conf. on Computer Vision and Pattern Recognition Workshop*, pages 1 – 7, 2008. 51, 61, 116

[122] B. J. Super and A. C. Bovik. Three-dimensional orientation from texture using gabor wavelets. In *Proc. SPIE Visual Communications and Image Processing '91: Image Processing*, 1991. 61, 64, 71, 72, 73

[123] B. J. Super and A. C. Bovik. Planar surface orientation from texture spatial frequencies. *Pattern Recognition*, 28(5):729–743, 1995. 61, 62, 63, 64, 65, 71, 72, 73, 89, 92, 97

[124] B. J. Super and A. C. Bovik. Shape from texture using local spectral moments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(4):333–343, 1995. 63, 64, 71, 72, 73

[125] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1 – 9, 2015. 35, 38

[126] G. Tsai, C. Xu, J. Liu, and B. Kuipers. Real-time indoor scene understanding using Bayesian filtering with motion cues. In *Proc. IEEE International Conf. on Computer Vision*, pages 121 – 128, 2011. 48

[127] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *Proc. European Conf. on Computer Vision*, pages 589–600, 2006. 19

[128] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008. 137

[129] A. Vedaldi and K. Lenc. Matconvnet — convolutional neural networks for matlab. In *Proc. ACM Int. Conf. on Multimedia*, 2015. `http://www.vlfeat.org/matconvnet/`. 145

[130] J. Vogel, A. Schwaninger, C. Wallraven, and H. H. Blthoff. Categorization of natural scenes: Local versus global information and the role of color. *ACM Trans. on Applied Perception*, 4(3), 2007. 15

[131] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. HOGgles: Visualizing object detection features. In *Proc. IEEE International Conf. on Computer Vision*, pages 1 – 8, 2013. 45

[132] A. Wang, J. Lu, G. Wang, J. Cai, and T.-J. Cham. Multi-modal unsupervised feature learning for RGB-D scene labeling. In *European Conference on Computer Vision Workshop on Applications of Computer Vision*, pages 453–467, 2014. 48

[133] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3360–3367, 2010. 13, 16, 24

[134] L. Wang, Y. Li, J. Jia, J. Sun, D. Wipf, and J. Rehg. Learning sparse covariance patterns for natural scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2767 – 2774, 2012. 19, 20, 26, 40

[135] X. Wang, T. Han, and S. Yan. An HOG-LBP human detector with partial occlusion handling. In *Proc. IEEE International Conf. on Computer Vision*, pages 32 – 39, 2009. 46

[136] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(2):210 – 227, 2008. 20

[137] J. Wu and J. M. Rehg. CENTRIST: A visual descriptor for scene categorization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(8):1489–1501, 2011. 24, 33, 138, 139

[138] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva. SUN Database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, pages 1–20, 2014. 10, 13, 24, 34, 40, 140, 142, 144

[139] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN Database: Large-scale scene recognition from abbey to zoo. In *Proc.*

IEEE Conf. on Computer Vision and Pattern Recognition, pages 3485 – 3492, 2010. 13, 20, 24, 27, 34, 38, 40, 44, 140, 142, 144

[140] L. Xie, J. Wang, B. Guo, B. Zhang, and Q. Tian. Orientational pyramid matching for recognizing indoor scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3734 – 3741, 2014. 15, 41, 42

[141] J. Yang, , K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1794–1801, 2009. 16, 18, 20, 23, 24, 25, 40

[142] J. Yang, K. Yu, and T. Huang. Supervised translation-invariant sparse coding. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3517 – 3524, 2010. 22

[143] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *Proc. IEEE International Conf. on Computer Vision*, pages 543 – 550, 2011. 20

[144] K. Yu, Y. Lin, and J. Lafferty. Learning image representations from the pixel level via hierarchical sparse coding. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1713–1720, 2011. 16

[145] L. Zelnik-Manor, K. Rosenblum, and Y. Eldar. Dictionary optimization for block-sparse representations. *IEEE Trans. on Signal Processing*, 60(5):2386 – 2395, 2012. 19

[146] C. Zhang, J. Liu, Q. Tian, C. Xu, H. Lu, and S. Ma. Image classification by non-negative sparse coding, low-rank and sparse decomposition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1673–1680, 2011. 17, 25, 40

[147] J. Zhang and T. Tan. Affine invariant classification and retrieval of texture images. *Pattern Recognition*, 36(3):657–664, 2003. 90, 165

[148] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, 1998. 70

[149] Z. Zhang, X. Liang, A. Ganesh, and Y. Ma. TILT: Transform invariant low-rank textures. In *Proc. Asian Conf. on Computer Vision*, pages 314–328, 2010. xiii, xiv, xv, xviii, 62, 63, 92, 94, 95, 96, 99, 103, 107, 108, 109, 110, 130, 131

[150] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva. Places: An image database for deep scene understanding. *arXiv preprint*, 2016. `http://places2.csail.mit.edu/`. ix, xv, xvi, 40, 136, 153, 154, 155, 158

[151] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using Places Database. In *Proc. Neural Information Processing Systems*, 2014. 40

[152] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2879 – 2886, 2012. 46

[153] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, and X. Jiang. Learning discriminative and shareable features for scene classification. In *Proc. European Conf. on Computer Vision*, pages 552–568, 2014. 39, 41, 42

And He taught Adam the names – all of them. Then He showed them to the angels and said, "Inform Me of the names of these, if you are truthful." They said, "Exalted are You; we have no knowledge except what You have taught us. Indeed, it is You who is the Knowing, the Wise." He said, "O Adam, inform them of their names." And when he had informed them of their names, He said, "Did I not tell you that I know the unseen (aspects) of the heavens and the earth?"... And (remember) when We said to the angels, "Prostrate before Adam"; so they prostrated, except for (the arrogant jinn) Iblees (Satan)... And We said, "O Adam, dwell, you and your wife, in Paradise and eat therefrom in abundance from wherever you will. But do not approach this tree, lest you be among the wrongdoers."...

Al Qur'an 2:31–35