UNIVERSITY OF SOUTHAMPTON

Semantics of Texture

by

Tim Matthews

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Physical Sciences and Engineering
School of Electronics and Computer Science

July 2016

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Tim Matthews

In this thesis we investigate means by which the semantic and visual spaces of texture may be tied together, and argue for the importance of explicit semantic modelling in human-centred texture analysis tasks such as retrieval, annotation, synthesis, and zero-shot learning.

We take a new approach to semantic texture labelling by adopting a pairwise comparison framework robust to human biases, and within a semantic space consisting of *attributes*, low-level visual features acting as building blocks for more expressive semantic ontologies. We crowdsource a dataset of approximately 140,000 pairwise comparisons across 319 classes of texture and 98 attributes — to our knowledge the largest of its kind.

To aid in learning from sparsely labelled pairwise comparison datasets such as this we derive a new Bayesian probabilistic approach, providing a natural framework in which to incorporate prior knowledge and to measure uncertainty, and outperforming the often-used Ranking SVM on incomplete and unreliable data. We demonstrate how the error variance present in our pairwise comparison data may be precisely quantified, allowing us to identify and discard rogue responses in a principled way.

Existing texture descriptors are then assessed in terms of their correspondence to the attributes comprising the semantic space. Textures with strong presence of attributes connoting randomness and complexity are shown to be poorly modelled by existing descriptors. These effects are likely due to disparities between human perception of what texture entails, and definitions adopted prior to (or, sometimes, after) the design of computational texture analysis systems.

Despite the deficiencies of the visual descriptors they are based upon, we demonstrate the benefit of semantically enriched descriptors in a retrieval experiment. Semantic modelling of texture is shown to provide considerable value in both feature selection and in analysis tasks.

Contents

List of Figures

List of Tables

Acknowledgements

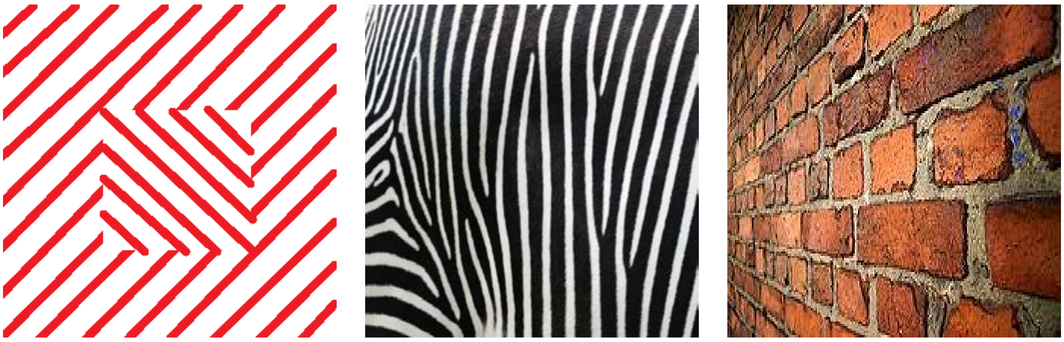
Many thanks to all members of the group who have assisted – directly or indirectly – with this work so far, in particular my supervisors Mark Nixon and Mahesan Niranjan for their guidance and inspiration, and whose students have been particularly helpful in many ways. I'd like to express my gratitude to each of them for expanding my academic horizons more than they would have been in the absence of their presentations and discussions.

I am particularly appreciative of the encouragement and motivation given to me from my family, my supportive girlfriend Nazli, and from Dr. Sarvapali Ramchurn, who gave useful feedback.

Chapter 1

Context and Contributions

Visual texture is an important cue in numerous processes of human cognition. It is known to be used in the separation of 'figure' from 'ground', as a prompt in object recognition, to infer shape and pose, as well as in many other aspects of scene understanding (see Figure 1.1). Over eons of human existence this importance has led to the development of a rich lexicon suitable for concise description of texture. We may speak of *fractured* earth, or of a *rippling* lake, and in doing so are able to convey considerable information about the surface and appearance of these objects.



(a) The *figure* (the centre 'square') is readily segmented from *ground* based only on the orientation of texture primitives.

(b) The depicted object is identifiable due to its distinctive texture.

(c) The pose of the wall is inferred from the perspective effect of its texture.

FIGURE 1.1: Three examples of texture's importance in human scene understanding.

Although computational texture analysis has achieved fine results over recent decades, there still remains a disparity between the visual and semantic spaces of texture – the so-called *semantic gap*. Computational approaches usually operate on the basis of *a priori* notions of texture not necessarily tied to human experience. This means they are often unsuitable for applications requiring closer or more intuitive human interaction.

The overall goal of this thesis is therefore to investigate the semantic and visual spaces of texture and to attempt to identify correspondences between these spaces that align well with human perception. We discuss real-world motivations for doing this in the next section, the challenges we will face in Section 1.2, and give an overview of the contributions we make to solving this problem in Section 1.3.

1.1 Motivation and applications

We pursue our goal primarily from the perspective of content-based image retrieval (CBIR). Systems of this type attempt to retrieve images in response to user queries, based only (or at least in part) on the visual content of those images. Typical CBIR systems operate on low-level features such as colour histograms and co-occurrence matrices (see Section 2.3.2.1) but this approach is often unsatisfactory for human users with a sophisticated understanding of the high-level concepts contained in the images. For this reason there has been a greater focus towards designing CBIR systems with explicit modelling of these high-level semantics (Liu et al., 2007). Separate semantic modelling has been shown to improve retrieval of natural scenes (Vogel and Schiele, 2007), gait signatures (Samangooei et al., 2008; Reid et al., 2014), and indoor-outdoor classification of photographs (Serrano et al., 2004).

CBIR is part of a more general class of problems known as *symbol grounding* problems, which have wide-ranging applications. We discuss the symbol grounding problem in more detail in Section 2.4. Solving the symbol grounding problem also opens up the possibility of *zero-shot learning* in classification tasks. Here, a classification system can be taught new categories without ever needing to observe them, through processing of some semantic description of the category. This is of particular use in robotics, where the environment is unpredictable and so it is advantageous to learn dynamically about possible interactions and scenarios without first having to encounter them (Palatucci et al., 2009).

Further applications lie in the related problems of texture synthesis and texture description. In the former of these, textures are computationally generated to conform to specifications communicated through semantic user input without the user requiring any knowledge about the specific underlying generating algorithms or their parameters. The inverse description problem involves the automatic annotation of texture images using forms borne from the expressive semantic space of texture. The latest `MPEG` standard – `MPEG-7` – mandates standards for the semantic description of visual multimedia and employs basic measures of regularity and coarseness as part of a rudimentary texture browsing descriptor (Manjunath et al., 2001). Its introduction can be interpreted as a response to growing consumer interest in the benefits such description schemes provide.

1.2 Challenges

The simplest form of symbol grounding – mapping pixel-wise image intensities to expressive high-level semantic forms – is challenging due to the dimensionality of the data in question and the complexity held within it. Instead, appropriate structuring of both the semantic and visual spaces is desirable. One common consideration in this structuring is in the problem of feature selection, the selection of succinct measures representative of visual (or semantic) qualities. Much of this thesis will be devoted to exploring possible forms these spaces may take in order to identify a balance between efficiency and expressiveness. The creation of CBIR systems also comes with its own specific challenges – for instance the design of similarity measures and efficient indexing and retrieval mechanisms – although we will not concern ourselves with these issues in this thesis.

Further challenges lie in replicating human visual performance. Humans are able to process and interpret complex scenes extremely accurately in fractions of a second, and very little is known about the exact mechanics of how this is done. On top of this, humans possess a remarkable ability to generalise – to learn new and unfamiliar visual stimuli quickly and reliably. Again, the details of how this is done are hazy. We state what we *do* know about the human cognitive processes for texture in Section 2.2, and may hope to take inspiration from these in an eventual computational implementation.

Texture in particular provides interesting challenges of its own, partly because it has historically proven so difficult to definitively define. We devote some attention to this issue in Section 2.1.

1.3 Contributions

The main contributions of this thesis are:

- A new approach to texture characterisation and analysis, tying together previous literature on the semantic and visual spaces of texture with a pairwise comparison and probabilistic learning procedure to model how texture is described by human subjects. We explore semantic modelling in general, as well as specific issues that arise when applying it to texture, and demonstrate excellent performance within this setting.
- A large publicly-available dataset of 140,000 labels of 319 classes of texture from the Outex dataset (Matthews, 2014). The design and format of the labelling is described in Chapter 3.
- A new probabilistic technique for learning item ratings from pairwise comparison data. This Bayesian technique allows users to incorporate prior knowledge into

the learning procedure and to precisely quantify the uncertainty surrounding the inferred item ratings. This is done in Chapter 4.

- An appraisal of how well a selection of visual texture descriptors are able to capture this semantic data, and a demonstration of the benefits of explicit semantic modelling for texture retrieval, both given in Chapter 5, and comprising a paper presented at CVPR-2013 (Matthews et al., 2013).

We finish with discussion of our results and an outline of our future intentions in Chapter 6. The next chapter begins with a review of relevant literature and background material.

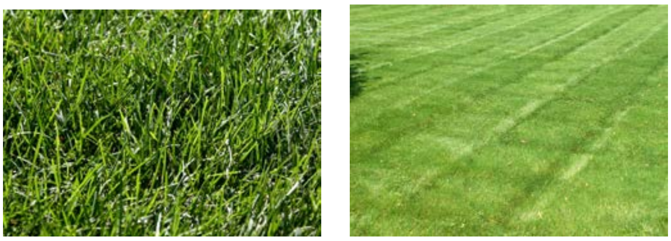
Chapter 2

Background and Literature Review

We begin this chapter by characterising what is meant by visual texture in more detail, starting with its definition in the next section, how it is used and understood by humans in Section 2.2, and how it is processed by computer systems in Section 2.3. We then discuss the symbol grounding problem and its application to texture in Section 2.4, before providing an overview of semantic spaces in Section 2.5.

2.1 What is texture?

In general usage "*texture*" describes the tactile sensation of a surface caused by its three-dimensional structure: the Chambers 21$^{\text{st}}$ Century Dictionary gives its primary definition of the word as "*the way the surface of a material or substance feels when touched*". For instance, the closely packed blades of grass of a mown lawn feel bristled, and touching sculpted marble provides a sensation of smoothness. When light incident on a surface is reflected to some observer, the resultant image contains visual information about this 3D surface structure. The human brain is adept at discerning surface properties based on this visual information due to – we may presume – the obvious practical advantages of not being required to individually touch the surfaces comprising an unknown scene in order to learn about it, and the ubiquity of texture within the natural world. We can introduce a tentative initial definition of *visual* texture as being those properties of an image region which convey tactile information about a corresponding surface. From this we note that visual texture is necessarily a scale-dependent image feature, it being a product of both the observer's visual resolution and viewing conditions as well as of the scale of the surface structure (see Figure 2.1).



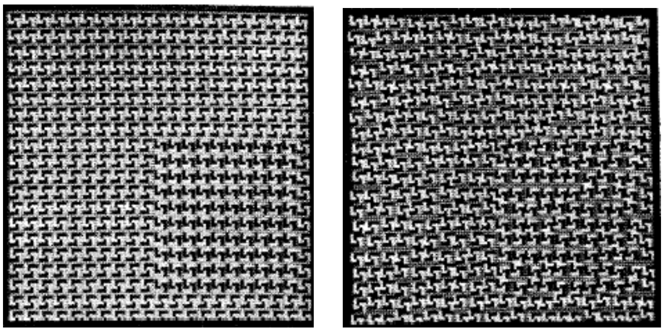
(a) Close-up view of mown grass, demonstrating an irregular, uneven appearance with repeated linear elements

(b) Elevated view of mown grass, revealing a regular striping pattern caused by the mowing technique

Figure 2.1: Scale-dependence of texture

At this point we may extend this initial definition to also include properties which convey *any* distinguishing information about a surface, not necessarily the result of 3D structure. The geological composition of marble causes its characteristic smudges and veins which – whilst not appreciable tactually – allow it to be rapidly recognised and distinguished by an observer. This extension introduces an issue of complexity, as even quite convoluted and non-obvious visual information may be used to distinguish a surface given sufficient attention by the observer. Indeed, pioneering work in Julesz (1962) and Julesz (1975) introduced various measures of texture complexity, and demonstrated a correspondence between these complexity measures and the ability of the preattentive visual system – the part of the visual system responsible for rapid, non-conscious scene understanding – to use them as cues in texture discrimination. An example of the role played by preattentive processing in texture segmentation is shown in Figure 2.2. Here, there is a clear difference between the ability of the preattentive visual system to segment the two images, but it is difficult to express specific properties of texture that affect this. Visual texture is therefore a context-dependent quality, with visual properties holding textural meaning in proportion to how rapidly they may be processed by the visual system and how useful they are for scene understanding once processed.

This definition is of course hazy and subjective to some observer. It is intended primarily for illustration of the notions of scale-dependence and complexity and of the importance of texture as a visual cue. It also allows us to stress the distinction between tactile texture and visual texture – throughout the remainder of this thesis "texture" given without specification will be in reference to the latter. Even within disciplines in which texture is of high importance – such as computer vision – it has proven difficult to obtain unified assent for a specific definition of texture. Objective definitions have been proposed based variously on generative rules and regional statistics (Tuceryan and Jain, 1998); or *a posteriori* based on dataset contents or the outputs of existing algorithms designed to detect or synthesise texture (Nixon and Aguado, 2012). Although a precise definition is desirable at this early stage due to its influence on dataset and feature selection (amongst other things), for our specific purposes a subjective definition of texture embedded in human experience may be appropriate. Because our task involves



(a) Image in which the bottom-right quadrant is preattentively distinguishable from the remaining three quadrants.

(b) Image in which the bottom-right quadrant requires a degree of conscious inspection in order to distinguish the bottom right quadrant, through random jittering of (a).

Figure 2.2: Images in which the bottom-right quadrant is the inverse of the other three quadrants. Taken from Julesz (1962).

tying some visual texture space to a semantic space borne from human interpretation of that visual space, it is fitting to adopt a definition of texture derived from human perception. In this sense texture is anything describable by constructions from our semantic space and emerges as a natural consequence of our eventual definition of that space.

It is also worth making specific reference to colour, the understanding of which is far more sophisticated than for any other visual cue. It is known that the entire gamut of human colour vision can be represented in three dimensions and that distances between the colours within this gamut can be calculated which exhibit good accordance to human perception. A cross-cultural study by Berlin and Kay (1969) provided evidence that, not only is colour understanding driven by innate rather than cultural processes, but also that the way language is used directly reflects the underlying cognitive structure. Furthermore, reasonably successful methods of mapping colour semantics onto this gamut have been devised (Tominaga, 1985; Mojsilovic, 2005). Despite the physical processes underlying colour being far simpler than that for texture, the field of colour research is well worth looking towards as a model for progress in understanding texture. Because of this relatively mature understanding, we consider colour to be a completely separate visual cue that can be used to augment an achromatic texture model at a future point. We thus only consider the achromatic properties of texture throughout this thesis.

In the next section we give an overview of how visual texture is processed and understood in biological visual systems.

2.2 Cognition of texture

As with most visual processing, human cognition of texture is complex and not fully understood. However, biologically-inspired machine analogs of vision are highly desirable due to the performance and flexibility of the human system (presumably due to its architecture and cognitive processes), as well as the intuition that visual cognition is intrinsically tied to language, and so understanding the nature of this cognition brings us some way closer to understanding the linguistic structure.

Unfortunately, the neurological underpinnings of textural processing within the visual cortices of the brain are difficult to decipher and attach functional meaning to. Study of the brain at this level is the domain of neurophysiology, but this field has obtained only limited explanatory success. Research can alternatively adopt a psychophysical methodology, treating the brain as a "black box" for which the behaviours are to be determined experimentally and then seeking plausible cognitive architectures which cohesively explain these empirical observations. Clearly, study can be approached from multiple directions, and this fact may be expressed in terms of the abstractional tiers of Marr (1982), those three levels of analysis necessary to gain full understanding of a visual system:

- *Computational* understanding: the overall goals of the system and its constraints.
- *Representational* understanding: the representations and algorithms used to manipulate and process visual texture.
- *Implementational* understanding: the physical architecture of the system.

In the remainder of this section we provide an overview of human cognition of texture in terms of these three levels of analysis.

2.2.1 Computational analysis

In humans, even the computational aspect of texture is unclear: it is not known definitively what texture is used for or how, at a high-level, it is processed. As mentioned previously, texture is known to play a fundamental role in segmentation – the partitioning of visual scenes into contiguous regions – as well as in the related problem of figure-ground separation, whereby objects of interest are given focus through preattentive separation from irrelevant background noise. Lamme et al. (1999) proposes two possible high-level logical processes for texture segmentation, one initiating from boundary detection followed by a texture-guided 'filling-in', and another based on clustering of visible surfaces using texture measures as similarity metrics. His experiments support the former of these hypotheses, suggesting that texture is an important *secondary* cue, gaining cognitive significance only after the detection of more fundamental visual features. This corroborates the hypothesis of Marr (1982) in which vision proceeds from a 2D "primal sketch" – formed from basic features such as edges – to a full 3D model only via an intermediate '2.5D' level, at which texture cues are used to flesh out the initial sketch.

2.2.2 Implementational analysis

The physical process of textural understanding commences when photons are reflected from some visual field onto the *retina*, an area of tissue at the back of the eye consisting of an array of approximately 130 million structures individually sensitive to the intensity and wavelength of light (Hubel, 1995). Wavelength-sensitive excitation of these structures is collated across the retina and eventually processed to create the sensation of colour. In accordance with our achromatic perspective of texture (as set out in Section 2.1) we ignore colour and restrict our attention to the corresponding intensity excitations. Intensity-sensitive responses collected across the retinal structures are subject to rudimentary preprocessing before undergoing spatial compression: the ≈130 million responses are mapped onto a far smaller number of *ganglion cells.* These ganglia comprise the *optic nerve*, a pathway from each eye transmitting the spatially compressed retinal information to the *optic chiasm* – responsible for extracting the left and right visual fields from both retinal images for subsequent binocular processing – and in turn to the visual centres of the brain. Each visual field is localised to one of the two brain hemispheres, with the left hemisphere handling the right visual field and vice versa: the remainder of this overview is hemisphere-specific (although both operate similarly upon their visual field).

The first centre to receive visual input is the *lateral geniculate nucleus*, which acts in part as a relaying hub to different areas of the brain. In particular, each lateral geniculate nucleus forwards input to a *visual cortex*, which may be loosely viewed as an umbrella over five interconnected child areas, labeled V1 to V5, built up of a connected network of *neurons*, each tuned to fire a signal to its neighbours in response to specific stimuli. These five areas roughly establish a pathway, with more complex representations of form and shape elaborated along this pathway. Of these areas, V4 is thought to be the most important in texture understanding (Hanazawa and Komatsu, 2001), with lesions in this area shown to inhibit discrimination between textures (Schiller, 1993; Merigan, 2000). Most knowledge has come from functional magnetic resonance imaging of macaque monkeys, known to have a similar brain structure to humans. It seems texture understanding proceeds in V4 through neuronal encoding of elements ranging in complexity from basic oriented lines (Desimone and Schein, 1987) to complex shapes, with the most complete picture due to Okazawa et al. (2015), who found V4 neurons appear to be tuned towards "*sparse combinations of higher order image statistics*", in particular those of Portilla and Simoncelli (2000) whose statistics comprise in part the outputs of Gabor-like filters at various locations, scales, and orientations.

2.2.3 Representational analysis

Early representational analysis of texture proceeded from definitions of texture based on the spatial arrangement of certain texture primitives. Julesz (1981) labelled these primitives *textons*, and considered them to be the atomic units of preattentive texture cognition, considered loosely to be small elements such as dots and lines. No real attempt was made to formalise these atomic units, or to explain their representational form, and Julesz's analysis was largely restricted to artifically synthesised textures. Work by Karni and Sagi (1991) suggested the internal texton representations within the brain were in fact adjustable through training, suggesting they are learnt from the primitive features in textures encountered through normal existence – so-called natural textures.

Other research has eschewed the notion of specific primitives. Harvey and Gervais (1981) subjected viewers to synthesised textures and found "*visual internal representation of stimuli is based on spatial frequency analysis rather than feature extraction*". Experimentation on macacques performed by Arcizet et al. (2008) determined banks of Gabor filters – spatially localised frequency detectors – to be good approximations to texture perception, and numerous models have operated under this assumption (Malik and Perona, 1990; Landy and Bergen, 1991).

These different approaches – based on primitives and frequency responses – are not necessarily conflicting. Primitives may be united cohesively with filtering processes by interpreting them to be the responsive peaks of these processes, as they may be in Gabor filter banks. Alternatively, any combination of representational forms may exist in conjunction, either in parallel, or as different levels of an overall pyramidal representation. Zhu et al. (2005) formalise the notion of textons and integrate them with oriented spatial frequency filter within a hierarchical generative model allowing dictionaries of textons to be learned and, from these texton dictionaries, dictionaries of filters to be derived.

Having given an overview of human texture processing, in the next section we give an outline of computational efforts at performing the same task.

2.3 Computational texture representations

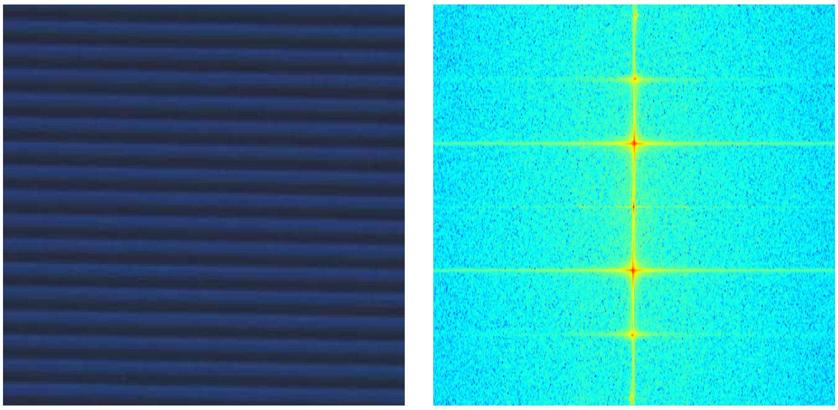
In this section we present six different feature space representations for visual texture, which will be referred to throughout this thesis. We look at two main categories of representation: decompositional approaches, which convert image data to some alternative representation before proceeding with a separate feature extraction procedure; and statistical, in which features are extracted directly from the intensities of the original image.

2.3.1 Decompositional representations

Decompositional approaches to feature extraction operate by decomposing the input image into some alternative representation, where this representation is chosen in the hope it sheds light on some aspect of texture. Features are then computed from these alternate representations.

2.3.1.1 Fourier transforms

The Fourier transform of an image is a representation of that image within the frequency domain. It derives from the theory of Fourier analysis, whereby it is shown that any signal can be decomposed into a sum of sinusoids of different frequencies. An example transform is shown in Figure 2.3. Fourier transforms are appealing for texture analysis due to their ability to capture information regarding the repetitive nature of texture. Highly repetitive texture can be expected to possess peaks in frequency. As such, we may expect Fourier-based methods to discriminate well between ordered and disordered textures, which we shall see in Section 3.2.2 occupy an important place in the semantic space of texture.



(a) Source image, demonstrating regular horizontal repetition.

(b) Amplitude component of the Fourier transform of the image, with the highest amplitudes aligned to the vertical image axes due to the source image's horizontal repetition.

FIGURE 2.3: Fourier transform of a highly repetitive image.

Feature selection for Fourier transforms should thus focus on measures able to succinctly capture the degree and form of repetition. Liu and Jernigan (1990) propose a set of 28

Fourier transform features, with an emphasis on good performance in noisy conditions. These are derived from the normalised Fourier transform F^N, obtained by dividing the source transform F by the root of its squared amplitudes:

$$F^N(u,v) = \frac{|F(u,v)|}{\sqrt{\sum_{u\neq 0, v\neq 0} F(u,v)^2}} \tag{2.1}$$

Identical normalised transforms are also calculated specific to the four quadrants of F, denoted $F_1^N, \ldots, F_4^N$. A subset of eight of the 29 Liu features, $f_1, \ldots, f_8$, are designated as being "optimal" descriptors and are listed below:

- 1) Normalised amplitude of the major peak:

$$f_1 = \max_{u,v} [F^N(u,v)] \tag{2.2}$$

- 2) Laplacian of the major peak. That is, the sum of the amplitudes of the major peak's four neighbours:

$$(u_1, v_1) = \arg\max_{u,v} [F^N(u,v)] \tag{2.3}$$

$$f_2 = F^N(u_1-1,v_1) + F^N(u_1+1,v_1) + F^N(u_1,v_1-1) + F^N(u_1,v_1+1) - 4f_1 \tag{2.4}$$

- 3) Squared major peak frequency:

$$f_3 = u_1^2 + v_1^2 \tag{2.5}$$

- 4) Inertia in first quadrant:

$$f_4 = \sum_{u>0,v>0} \sqrt{u^2+v^2} F^N(u,v) \tag{2.6}$$

- 5 & 6) Total amplitude of first two quadrants:

$$f_5 = \sum_{u>0,v>0} F^N(u,v) \tag{2.7}$$

$$f_6 = \sum_{u<0,v>0} F^N(u,v) \tag{2.8}$$

- 7 & 8) Relative entropy of first two quadrants:

$$f_7 = \frac{\sum_{u,v} F_1^N(u,v) \ln F_1^N(u,v)}{\sum_{u,v} \mathrm{I}\,[F_1^N(u,v) > 0]} \tag{2.9}$$

$$f_8 = \frac{\sum_{u,v} F_2^N(u,v) \ln F_2^N(u,v)}{\sum_{u,v} \mathrm{I}\,[F_2^N(u,v) > 0]} \tag{2.10}$$

where I denotes the $0 - 1$ indicator function.

A special case of the Fourier transform is described in the next section: the Gabor transform.

2.3.1.2 Gabor transforms

A Gabor transform is a special case of a Fourier transforms localised to a specific spatial area using a Gaussian window. Multiple Gabor filters are typically used in combination, tuned to different orientations and scales. An example of a combination of Gabor filter responses is shown in Figure 2.4: because the filters are not orthogonal, they are spaced such that their half-peak responses touch. Banks of Gabor filters formed in this way have been reported to bear functional similarities to the primary visual cortex of the human brain (Jones and Palmer, 1987). Manjunath and Ma (1996) use the mean and standard deviation of each individual Gabor filter response as elements in a texture feature vector.

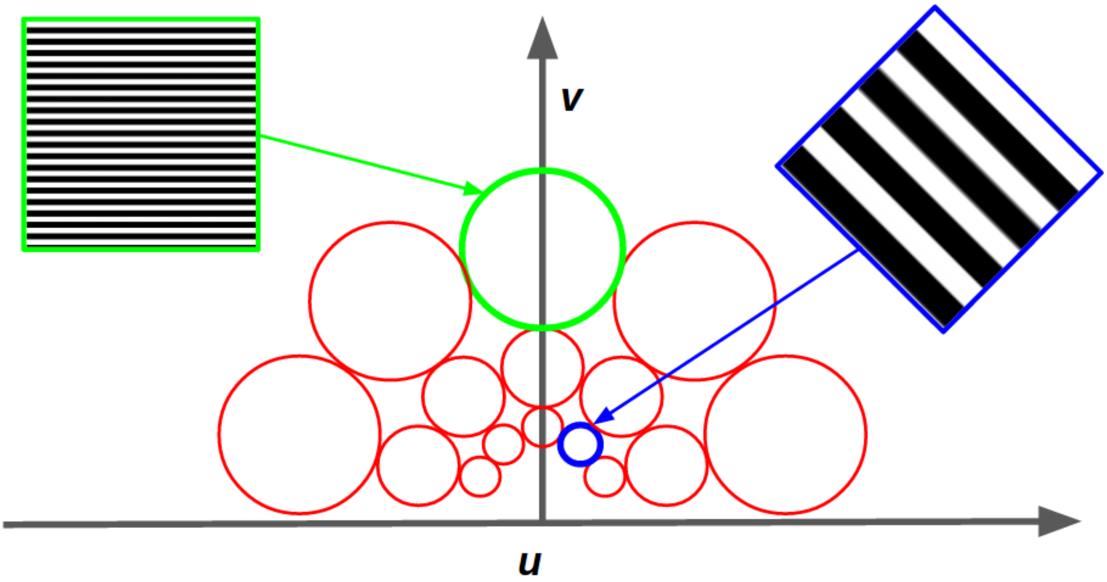


Figure 2.4: A combination of Gabor filters distributed over five orientations and at three different scales (a total of 15 filters) across two dimensions of spatial frequency, u and v. The circle perimeter represents the half-peak response of a single filter. Example textures delivering a high response for two of the filters are shown.

2.3.1.3 Statistical geometrical features

This approach was devised by Chen et al. (1995), and proceeds by decomposing an image into a stack of binary images by thresholding at regularly spaced intensity intervals. In the case of an 8-bit grayscale representation the decomposition process is lossless, as each of the 256 intensity levels can be thresholded to form corresponding binary images which

– when summed – combine to recreate the original image. An example decomposition for an image with four intensity levels is shown in Figure 2.5.

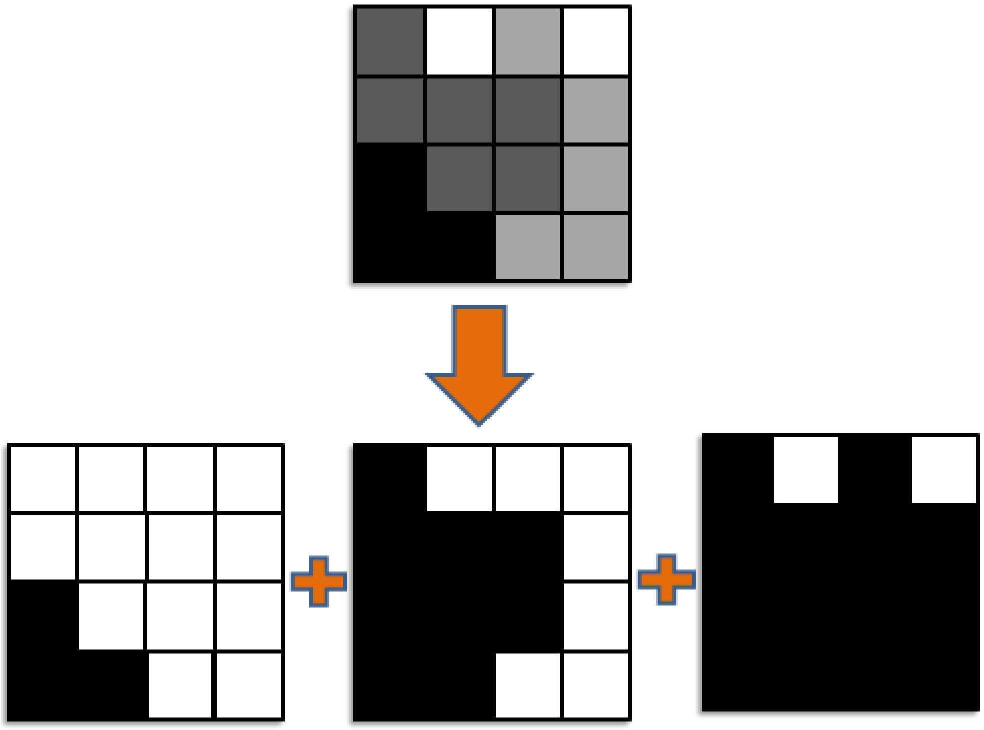


FIGURE 2.5: Decomposition of an image into three binary images by intensity thresholding.

Statistics are then calculated based on the number and shape of blobs across the range of binary images. The following quantities are obtained for each binary image:

- The number of white regions.
- The number of black regions.
- Average *irregularity* across white regions, where irregularity of a region is defined as the ratio between the largest distance between region pixels and the square root of the region area.
- Average irregularity across black regions.

Finally, the maximum, mean, sample mean, and sample standard deviation of each of these four measures are calculated across *all* binary images and combined to form a sixteen-dimensional feature vector.

Having described three decompositional texture representations, in the next section we describe three statistical alternatives.

2.3.2 Statistical representations

As opposed to decompositional approaches, statistical approaches of texture analysis operate directly on image intensity values, attempting to find patterns of localised grayscale structure characteristic to individual textures.

2.3.2.1 Co-occurrence matrices

Introduced by Haralick (1979), the co-occurrence matrix is a two-dimensional histogram of how often pairs of pixel intensities co-occur at a given distance and orientation from each other. For instance, a co-occurrence matrix may be used to count how many times an intensity of 244 occurred one pixel to the right of an intensity of 202. Clearly, for an 8-bit grayscale image the fully specified co-occurrence matrix will contain $256 \times 256 =$ 65,536 elements and will be mostly sparse, and for this reason it is often desirable to quantise the image intensities in some way. An example co-occurrence matrix for a 4x4 image quantised to contain only four intensity levels is shown in Figure 2.6.

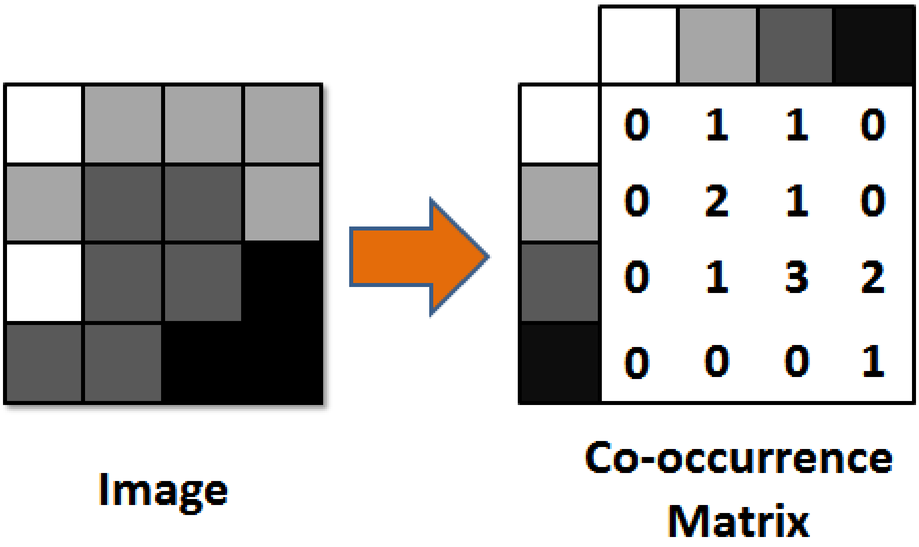


Figure 2.6: Example of a co-occurrence matrix counting intensity co-occurrences at a distance of one pixel in the direction of the positive x-axis.

Unless quantisation is performed co-occurrence matrices are generally too high-dimensional to be used as feature vectors themselves. Instead the matrix is normalised by dividing each element by the sum over all elements such that it may be interpreted as a two-dimensional discrete probability distribution. Then, various statistical measures are obtained from the weighted sum of these normalised matrix elements. Using p_{ij} to represent the normalised co-occurrence between the i^{th} and j^{th} intensities of an image with

n total intensity levels, the statistical measure m can be represented as:

$$m = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} w_{ij} \tag{2.11}$$

where w_{ij} is an element-specific weighting towards m, and m is interpreted in different ways depending on the values of $\boldsymbol{w}$. Some common interpretations are given in Table 2.1 below. Many of these are appealing in their intuitive meaning. Caution should be exercised in the weight placed on these interpretations however, as studies such as that of Tamura et al. (1978) have indicated only low correspondence between, say, *contrast* as defined in the table, and ordinary usage of the term as applied to visual texture.

Interpretation of m	w_{ij}
Dissimilarity	$\lvert i-j \rvert$
Contrast (or Inertia)	$\lvert i-j \rvert^2$
Homogeneity	$\frac{1}{1+\lvert i-j \rvert}$
Uniformity (or Energy)	p_{ij}
Entropy	$-\ln p_{ij}$
i mean, μ_i	i
j mean, μ_j	j
i variance, σ_i^2	$(i-\mu_i)^2$
j variance, σ_j^2	$(j-\mu_j)^2$
Correlation	$\frac{(i-\mu_i)(j-\mu_j)}{\sigma_i \sigma_j}$

Table 2.1: Common statistics computed from normalised co-occurrence matrices.

2.3.2.2 Local binary patterns

Local Binary Patterns are a technique for texture analysis introduced in Ojala et al. (1996) which identifies the histogram of the frequency of binary codings formed by taking focus pixels, and then thresholding pixels at a fixed distance from the focus based on its intensity. An example of this is shown in Figure 2.7 for eight points spaced regularly around the perimeter of a circle centred at the focus pixel. As eight perimeter points are used, the total number of possible codes is $2^8 = 256$.

Further study in Ojala et al. (2002b) identified specific patterns with the greatest discriminative power, named *uniform* binary patterns. These patterns had in common that they consisted of only two contiguous regions of lighter or darker pixels. It so happens that the local binary pattern depicted in Figure 2.7 is also a uniform binary pattern. Patterns of this type have appealing intuitive interpretations, as they can be seen to represent points, edges, corners, and line terminations – similar in many respects to the primitive textons of Julesz. The uniform binary patterns corresponded to successive powers of 2 – $2^0, \ldots, 2^8$ – in the original codings, and can be trivially made to be

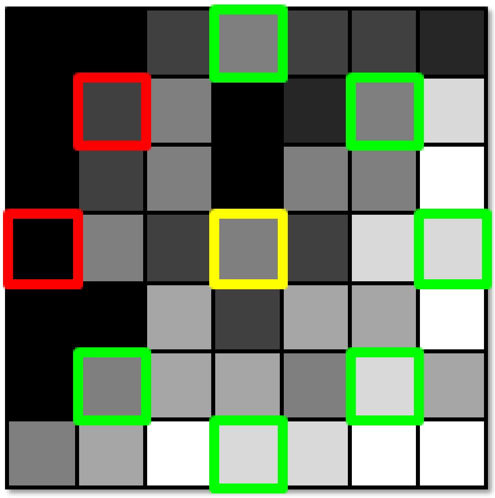


Figure 2.7: Example Local Binary Pattern yielding the code 11111100_2 (where 1 corresponds to green, and 0 to red) based on the intensity of pixels relative to the central yellow square.

invariant to rotation. These (normalised) histograms may be used as feature vectors directly: for n points selected around the circle perimeter, $n + 1$ uniform binary patterns are present. A further histogram bucket is often defined to count the number of non-uniform binary patterns, giving a final feature vector of size $n + 2$.

Interestingly, codings formed in this manner have found success elsewhere in computer vision – seemingly independently – as the basis for the FAST feature detector of Rosten and Drummond (2006), where they have been selected due to their robust properties in stereo correspondence problems and the speed in which they may be calculated.

2.3.2.3 Tamura texture features

Tamura et al. (1978) presented a set of six empirically selected texture attributes for which computational forms were derived. Although reasonably unprincipled in design and surpassed in intervening years by more sophisticated methods of texture analysis, Tamura et al.'s methods are interesting due to similarities with our own work. They share an attribute-based approach in common with our own, and a similar method of pairwise comparison was used to demonstrate good correlation between computational and human measures of attribute intensity for textures taken from the Brodatz dataset.

In the next section *symbol grounding* is discussed, it being at the heart of the problems tackled in this thesis.

2.4 The symbol grounding problem

Symbol grounding refers to the association of external meaning with the internal symbol set of a cognitive system. The problem of how this association is achieved in both biological beings and computational models has received considerable philosophical and psychological attention, and touches on fundamental aspects of cognition, computation, and consciousness. Artificial intelligence has often contributed and looked toward discourse on symbol grounding in its pursuit of models exhibiting intelligent behaviour, and various subfields assign their own name to specific forms of the problem: in the context of a robotic entity attaching meaning to sensory input it may be referred to as the "*anchoring problem*" (Coradeschi and Saffiotti, 2003), or as the "*semantic gap problem*" when associating query input with entries within some dataset, as in CBIR.

In a seminal paper on the topic, Harnad (1990) asks "*How can the semantic interpretation of a formal symbol system be made intrinsic to the system, rather than just parasitic on the meanings in our heads?*". Here, a *formal symbol system* is a system in which symbolic tokens are manipulated according to well-defined syntactic rules, and which is *semantically interpretable*, such that the state of the system and its constituent symbols at any point in time has intrinsic meaning. But can this meaning be inherent to the system at hand, instead of due to some external interpreter? It is not enough to encode the meanings of the symbols in some higher system as this encoding in turn must be composed of symbols, so leading to an infinite regress. Harnad instead argues that grounding must proceed from non-symbolic representations: an *iconic representation* consisting of projections of raw sensory data, and from this a *categorical representation* consisting of those invariant features of the iconic representation that have been determined to distinguish a (symbolic) category from others. From this perspective, symbol grounding is then the problem of how categorical associations are formed and maintained between continuous sensor data and discrete symbols sharing the same physical referent, so-called *categorical perception* (Harnad, 1987).

It is informative to relate the problem addressed in this thesis to the more fundamental problem of symbol grounding, and in particular to Harnad's representations described above. For a texture understanding system the iconic representation arises in the form of a feature space processed from raw signal data, examples of which were outlined in the previous section. The exact mechanism by which these iconic representations are distinguished and invariantly associated with symbols is a subject of active debate, but inextricably related to the underlying symbol system and the rules governing it. Computational models of categorical perception have found success, but address a far vaster problem than ours as they are often concerned with learning these representations from scratch. Instead we concern ourselves with just the interface between pre-defined iconic and symbolic representations, by adopting a few key assumptions about the symbolic representation, which we discuss below.

Firstly, we are sure the symbolic representation must consist in part of elementary base symbols corresponding to the invariant feature sets distinguished in the categorical representation. These base symbols are manipulated to establish higher level symbols, suggesting a hierarchical structure. Hierarchical structures have been commonly employed in computational visual systems, in which machine learning algorithms operate at the interface between these elementary symbols and visual feature spaces to learn the features distinguishing them. An example of a multi-level structuring adopted by Hudelot et al. (2005) is given in Figure 2.8, whereby the iconic representation is given in terms of spatially-localised statistics of the raw data, and the symbolic represention is an *a priori* defined concept ontology. Belkhatir (2005) uses a similarly structured approach specific to texture, in which primitive lexical units such as `bumpy` are tied to features detected using a bank of frequency filters.

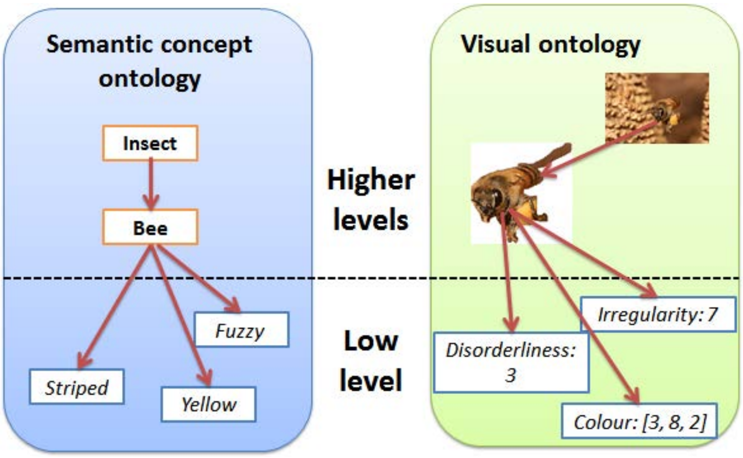


FIGURE 2.8: Multi-level hierarchy of Hudelot et al. (2005). The disparity between the two spaces is the semantic gap. The authors aim to close this gap by uniting the two lowest levels.

Next, how can we determine the elementary symbols? The approaches just described have in common that their symbolic representations are based on language. But do these computational systems employ language simply for ease of interpretation, or can language be seen as a reflection of the internal structure of the formal symbol system that is our mind? Certainly our mind's categorical representations are driven at least partly through language: were it the case that they were established purely through sensorimotor interaction, it would not be possible to be able to represent and distinguish new categories that have not been directly sensed. In computational models the communication of symbolic categories between entities ("*symbolic theft*") has been demonstrated to be more effective for learning categories than interacting with the physical environment ("*sensorimotor toil*") (Cangelosi and Harnad, 2001), and Vogt (2002) presents an experiment in which mobile robots establish a grounded lexicon through simple language games, stating "*language through its conventions offers a basis for invariant labeling of the real world*". There is also evidence that language is not only important for social knowledge transfer, but also plays a vital cognitive role in its own right through one's internal monologue (Jackendoff, 1996; Carruthers, 2002).

Our task then is in learning associations between feature space representations of texture and linguistic units relating to texture, which we take to be representative of the lowest levels of an underlying symbolic hierarchy, the grounded symbols from which higher level cognition proceeds. We refer to this shared structure between symbols and language as a *semantic space*, and explore different forms of semantic space in the next section.

2.5 Semantic spaces

In this section we specify how we may describe texture data in a flexible and expressive way, whilst simultaneously accounting for experimental and algorithmic feasibility. There is of course no one-to-one mapping between the visual and semantic spaces of texture. We demonstrate this in Figure 2.9, in which multiple forms of semantic description are given of the depicted texture, each with varying connotations, applicability, and degrees of expressiveness. These broad categories of semantic form are outlined below (and correspond to those given in the figure):

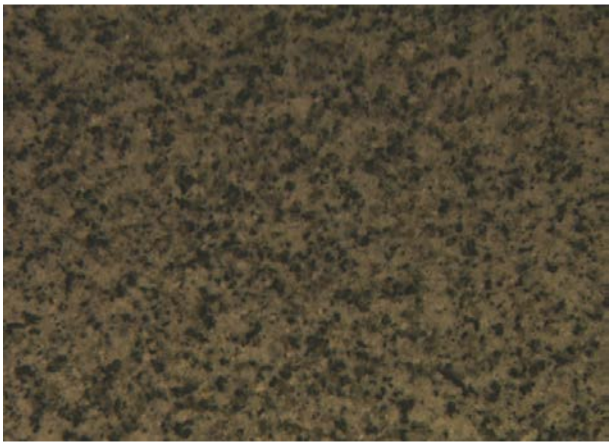


Figure 2.9: Visual texture with various descriptions corresponding to the semantic forms given in the text: a) *Granite*; b) *Like marble*; c) *Small smudged blobs distributed randomly across a lighter surface*; d) *Blemished, blotchy, random.*

- a) **Denotational** – the physical object or surface the texture corresponds to.
- b) **Analogy** – by reference to some other known category with similar appearance.
- c) **Visual** – description of the texture through reference to fundamental visual elements such as shape, and nature of repetition.
- d) **Attributes** – individual adjectives, the combination of which express the overall appearance of some texture (as discussed in Section 2.5.1).

Across these categories we note the recurrence of abstractional layers, in keeping with the multi-level view of symbol grounding introduced in Section 2.4. In particular, the **denotational**, **visual**, and **attribute** forms can be considered to represent decreasing levels of abstraction. Attributes can thus be seen as occupying the lowest layer of abstraction of all the forms listed above. In using them as building blocks for the other higher-level forms we strip away much of the linguistic and ontological complexity, and can consider the alleviation of these complexities as independent problems whose solution would eventually give rise to a fully-fledged semantic description system. In the next section we discuss attributes and their advantages in more detail.

2.5.1 Attributes

Attributes in computer vision denote low-level visual qualities – often adjectives – shared between objects, and they have received much attention in recent literature in the field, particularly within object recognition. Their use permits a shift in perspective from the traditional approach of object recognition in which object classes themselves are atomic units of recognition. They allow the association of visual data with shared low-level qualities, facilitating efficient class-level learning and generalisation, and they provide a means for intuitive and fine-grained description, such as when describing unusual features of an object, or in stating the ways in which one object is similar to another. Farhadi et al. (2009) state that attributes allow us to "*shift the goal of recognition from naming to describing*".

Because attributes are shared across objects in this manner they naturally may vary in their degree of expression across these objects, and so are amenable to flexible expression along some continuum, rather than in a binary fashion as is traditionally the case for category names. Consider the expressiveness of the attribute '*size of eyes*' as a continuous descriptor to be applied to cats, moles, and tarsiers (see Figure 2.10), as opposed to its binary equivalent '*has-eyes*', or – in the naming case – the broad and intrinsically binary categories of *cat*, *mole*, and *tarsier* themselves. It is clear that normal binary classification is insufficient for such attributes, as they correspond specifically to the *degree* of expression of visual features rather than to simply their presence or absence. Attributes have therefore found use in domains in which key features exist along continua, such as biometrics (Kumar et al., 2009; Reid and Nixon, 2011; Samangooei et al., 2008) and scene classification (Oliva et al., 1999; Rogowitz et al., 1998). Similarly, we see texture as being ill-suited to strict categorisation: key properties in which texture has been stated to vary include its *coarseness*, *linearity*, and *regularity* (Laws, 1980; Tamura et al., 1978), all of which are intuitively continuous qualities.

Figure 2.10: A tarsier.

2.5.2 Relative attributes

The continuous nature of many attributes has important consequences during data elicitation, in which human-provided labels are collected for each attribute in order to learn which visual features they align best with.

One common approach seen in the literature is to have subjects rate the perceived strength of attributes within each texture along a bounded rating scale, as is done for texture in Rao and Lohse (1993). Such scales typically require the respondent to select one of a number of rating levels, which labellers must reconcile with their own subjective interpretations — the scale cannot be *grounded* objectively, and different respondents may have different interpretations of what, say, '*average eye size*' entails. The inherent lack of grounding introduces inconsistency between respondents, and leaves them prone to numerous error factors — they may avoid extremal answers and be swayed towards *anchor points* along the scale (Chapman and Johnson, 2002), among other sources of bias (Tversky and Kahneman, 1974). Although careful experimental design can to some extent alleviate these flaws (Samangooei et al., 2008), the entire notion of a bounded continuum may still be inappropriate: the implication, for example, that there exists a *maximally* marbled texture is unreasonable and unintuitive.

An alternative technique involves having subjects hierarchically cluster items according to their similarity – as has been done for texture words by Bhushan et al. (1997). This approach removes many of the cognitive biases associated with rating scales, but hierarchical clustering places a greater cognitive burden on the respondent, typically requiring them to process the entire corpus of items for each attribute, which can be time-consuming and error-prone.

Instead, throughout this thesis, we operate within the framework of *pairwise comparison*, a psychometric procedure in which a subject is shown two stimuli simultaneously and prompted to choose the stimulus exhibiting more of some quality. Each pairwise comparison represents a snapshot of the underlying continuum for a given attribute, with inference proceeding using the *differences* between item features, rather than from the features themselves. In this way we work with so-called *relative attributes*, rendering the details of the underlying continuum immaterial to the respondent, who is concerned only with the binary decision of choosing which of two items exhibits more of the specified attribute. This is a cognitively simpler task which frees the subject from many of the aforementioned biases associated with other labelling schemes. Ultimately, a *ranking function* is learned, used in order to obtain attribute ratings for new unseen items. We will make use of pairwise comparisons, relative attributes, and ranking functions throughout this thesis.

Working with relative attributes offers numerous advantages, and the pairwise comparison setting is critical to our crowdsourced texture dataset described in Chapter 3, and to the specialised learning techniques we subsequently apply to this dataset. Relative attributes and pairwise comparisons have been shown to better reflect respondent beliefs than ordinary categorical attributes in retrieval tasks (Parikh and Grauman, 2011; Reid and Nixon, 2011) and to provide a more intuitive and natural user experience (Kovashka et al., 2012). They allow for greater nuance over traditional attributes, catering for forms such as '*animals with eyes larger than a tarsier but smaller than an owl*' that would not be otherwise expressible.

We mentioned earlier the impressive capacity for the human brain to generalise and to learn new categories of objects. Relative attributes represent a promising vehicle for between-class knowledge transfer – crucial in generalisation – and encouraging early results have been obtained when using them in such a way (Lampert et al., 2009; Farhadi et al., 2010). Taking this to the extreme, attributes facilitate *zero-shot learning* of completely unseen object categories through comparison with other categories that have already been learned (Wang et al., 2010). Parikh and Grauman (2011) present a neat way of doing this by calculating per-attribute rankings, mapping a semantic description on to them, and then deriving a newly learned feature vector corresponding to this description using the known features of similarly ranked items.

Pairwise comparison has previously been used specifically for texture by Tamura et al. (1978) across six attributes, but never on a large scale in a modern setting. There is evidence that the pairwise comparison scheme is particularly well-suited for texture: it has been hypothesised that the textural processes of the human cognitive system operate using such a comparison mechanism (Harvey and Gervais, 1981).

2.6 Summary

Much of what has been discussed in this chapter acts as a framework for the rest of the material in this thesis. We have looked in detail at texture – its definition and cognition –and at symbol grounding and appropriate structurings of the visual and semantic spaces. Flexible, expressive, low-level semantic units known as attributes will be heavily employed in coming chapters as we create a new dataset of texture words, formed using a cognitively appealing pairwise comparison procedure. As mentioned, little is known about how humans process visual texture or map it to the rich lexicon of words describing it, and so we must identify novel ways of learning from pairwise comparison data, seeking inspiration from data-rich domains such as machine learning and information retrieval.

We continue in the next chapter with the creation of a new crowdsourced dataset of relative attributes.

Chapter 3

Representations of Texture

In the last chapter we introduced the notion of a multi-level semantic and visual hierarchy (see Figure 2.8), where we may attempt to bridge the semantic gap by tying the two lowest levels of this hierarchy. In this chapter we give an overview of possible representations of these visual and semantic spaces, finishing by conducting a large-scale labelling task resulting in 140,000 pairwise comparison labels — one of the largest datasets of its kind.

We commence by detailing the dataset of visual texture we will use throughout this thesis.

3.1 Texture datasets

In selecting a dataset, it is useful to refer back to our characterisations of texture given in Section 2.1. Here, texture was presented as a nebulous visual quality, only loosely corresponding to the tactile sensation of physical surfaces. The importance of texture in human understanding of natural scenes was emphasised, as well as how this importance has given rise to the expressive semantic space which is the subject of this thesis. As such, our chosen dataset should span a diverse selection of textures, not simply limited to fabrics, abstract patterns, or any single domain in particular.

One of the largest and most widely-used datasets in texture analysis is the Outex dataset (Ojala et al., 2002a). The samples within this dataset vary in rotation, illumination, and scale. We assume texture perception to be both rotation and illumination-invariant, and although it is commonly held that visual perception of texture in humans is scale-invariant (Kingdom and Keeble, 1999; Zhang and Tan, 2002) we believe that it does not necessarily follow that this invariance exists within the semantic space: consider, for instance, how disordered circular primitives may appear either speckled or bumpy when observed respectively from afar or up close.

The size of the dataset, its widespread use, and its use of multiple samples of each texture are appealing, and we use the 319 texture classes included in test suite `Outex_TC_00016` throughout the rest of this thesis. We use samples within each class varying in illuminant (`horizon`, `inca`, `tl84`), and rotation (0°, 30°, 60°, 90°), but not in scale (only 100dpi is used). This gives twelve samples per texture class, for a total of 3828 samples all together.

Colour is taken to be a separate visual cue, and so all texture samples are converted to grayscale before experimentation. Examples of Outex textures can be seen in Figure 3.1.

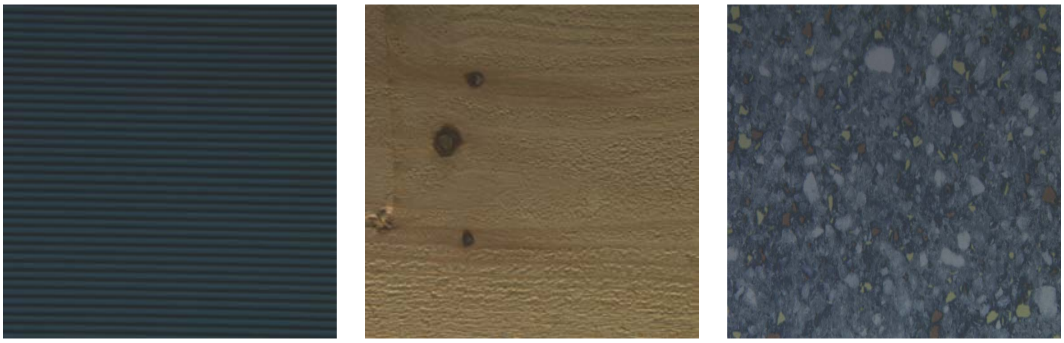


Figure 3.1: Examples of Outex textures.

Having set out the exact source of our visual space of texture, in the next section we describe the semantic space to which we will aim to tie it.

3.2 Semantic space of texture

Unlike for visual texture, where feature-space representations are well-studied and can be calculated directly from raw image data, the semantic space of texture is far more inscrutable, its structure having to be teased out through careful linguistic studies and psychometric surveys. Our aim is to find low-level semantic units — *attributes*, to use the terminology introduced in Section 2.5.1 — which will collectively make up a compact lexicon that is representative of how humans describe texture. In the next section we define the desirable qualities such a lexicon must have, before introducing the important study of Bhushan et al. (1997) who create an appealing texture lexicon which will prove fundamental to the rest of this thesis.

3.2.1 Lexicon selection

In defining a lexicon of attributes a trade-off must be made between two conflicting lexical qualities: *expressiveness*, and *efficiency*. The lexicon must be sufficiently expressive to be able to capture the variability of the visual texture space and, naturally, increasing the size of the lexicon will aid in achieving this goal due to the nuances of meaning connoted by the newly added terms. However the dimensionality of the lexicon has a direct influence on the complexity of algorithms involved in the eventual solution, and is known to play an important role in the accuracy of information retrieval systems (Ceglarek et al., 2010). As such we also wish to pursue the opposing goal of minimising its size – making it more efficient. Our focus, therefore, is on selecting a set of attributes which maximises some measure of expressivity, whilst remaining as small as possible.

Building up such a lexicon in an automated way is challenging, but the emergence of image-sharing sites such as Flickr featuring extensive user-submitted databases of tags – or *folksonomies* – has provided a rich source of descriptive data usable as a basis for knowledge extraction (Van Damme et al., 2007). Furthermore, some success has been found in work such as that of Neviarouskaya et al. (2011) in expanding sentiment word lists using the lexical ontology of WordNet (Fellbaum, 1998). Many authors such as Tamura et al. (1978) consider a manually selected attribute set based on their own empirical analysis of what appears to constitute texture. We may look to take a more principled approach, selecting words according to actual usage through the assumption that linguistic evolution has to a good extent balanced the expressiveness–efficiency trade-off for us. Fields such as food technology and petrology in which texture is of prime importance have developed mature, articulate lexicons to aid in their work. In the latter, for instance, *vesicular* describes a rock "*being pitted with many cavities at its surface*", but is a term – like many of the other terms used in such specialised fields – likely too unfamiliar to general audiences.

We have implicitly assumed our lexicon to consist of words from the English language. Many dozen WordNets have been constructed for languages other than English[1], and languages represented within the WordNet framework feature the advantage that standard tools – such as similarity measures – may be readily applied to them. It is also worth noting that a multilingual analysis can shed light on the underlying cognitive representation. A seminal study into colour semantics by Berlin and Kay (1969) across twenty languages revealed an intrinsic cognitive hierarchy of eleven categories. We leave the study of multilingual texture semantics for future work.

Of particular interest throughout this thesis is the lexicon compiled in Bhushan et al. (1997) using dictionary searching followed by a pruning process. The resulting 98 word lexicon reflected the semantic space of texture well enough to be able to account for 82%

[1] A listing is available at `http://www.globalwordnet.org/gwa/wordnet_table.html`

of the total variability of their experimental data. The methodology is specifically based around human textural understanding, and they are able to make numerous interesting characterisations of the semantic space of texture. We devote more attention to this study in the next section.

3.2.2 Bhushan et al.'s "Texure Lexicon"

To create their lexicon, Bhushan et al. used the following procedure. Initially, a 367 word lexicon of texture words was created through dictionary searching. Words were selected relating to *texture* or *patterns* and by examining synonyms of previously selected words. This was eventually pared down to a much smaller size by omitting rare and obscure words and words relating to light effects (e.g. *transparent*) to yield a lexicon of 141 words. These words were then presented to 20 subjects in a pilot study. Subjects were encouraged to form clusters of the 141 texture words based on their perceived semantic similarity. A similarity matrix was created based on how often pairs of words were clustered together by the respondents. A principal components analysis was performed on the results in order to remove insufficiently expressive words. The first seven principal components explained 70% of the variance in the data, and words without a high score in at least one of these components were removed from the lexicon as these words lacked the power to usefully describe different textures. This left a final lexicon of 98 words.

Next, the clustering task was performed again in a larger study involving 40 participants, and another similarity matrix was constructed, this time of size 98×98. Two important insights into the space of texture words were made using this similarity matrix:

- First, it was used to perform hierarchical clustering of the texture words, revealing eleven distinct clusters. We list interpretations of the eleven clusters in Table 3.1.
- Secondly, the authors performed multidimensional scaling on the similarity matrix and determined that three dimensions of visual texture were sufficient to explain 82% of variability in the collected similarity data. Only 59% of total variability was explained in the two dimensional case, while moving to four dimensions yielded only a 4% improvement. For this reason the authors focused on the three-dimensional structure. To better illustrate the structure of the space we plot the three-dimensional locations of a sample word from each of the eleven clusters in Figure 3.2.

The authors then confirmed a strong correlation between the semantic and visual spaces of texture through reference to the results of Rao and Lohse (1993), who performed a similar study restricted to the *visual* space of texture. The intuitive properties of this 98 word texture lexicon make it appealing for use as a source of low-level attributes in bridging the semantic gap. In particular, more modern dimensionality reduction

Cluster interpretation	**Sample words**
Linear orientation	*furrowed*, *lined*, *pleated*
Circular orientation	*coiled*, *flowing*, *spiralled*
Weave-like structure	*cross-hatched*, *meshed*, *woven*
Well-ordered	*regular*, *repetitive*, *uniform*
Disordered	*jumbled*, *random*, *scrambled*
Disordered linear primitives	*cracked*, *crinkled*, *wrinkled*
Disordered circular primitives	*dotted*, *speckled*, *spotted*
Disordered circular 3D primitives	*bubbly*, *bumpy*, *pitted*
Disordered weave-like structure	*frilly*, *gauzy*, *webbed*
Disordered indistinct circular primitives	*blemished*, *blotchy*, *smudged*
Disordered indistinct linear primitives	*marbled*, *scaly*, *veiny*

Table 3.1: Interpretations of the eleven texture word clusters identified in Bhushan et al. (1997)

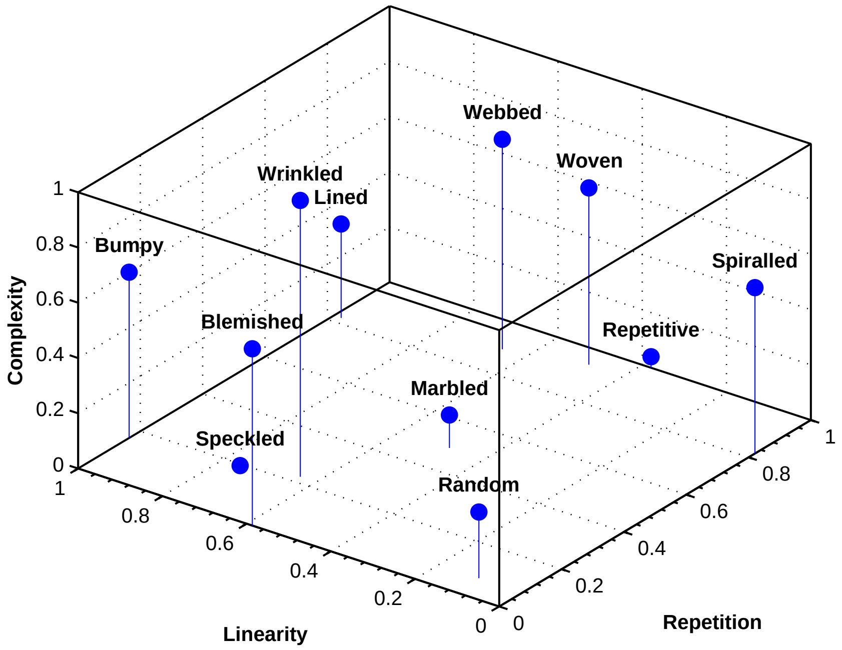


Figure 3.2: Representative attributes from each of the clusters in Table 3.1, with approximate locations (normalised between 0 and 1) across the three texture dimensions identified in Bhushan et al. (1997).

techniques could be applied in order to determine the most expressive subset of the lexicon of a given size, although unfortunately detailed results for the dataset are no longer available[2].

It is also worth emphasising the neat methodology employed by the study: the primary goal – maximising the expressiveness of a lexicon for texture – is too complex to elicit

[2] Established through e-mail correspondence with Rao.

from subjects directly, and they are instead only ever required to cluster the words based on perceived similarity. Clustering is a process essentially involving a series of binary comparisons, a particularly natural methodology for both experimenters and respondents, requiring less design input from the former and reducing the cognitive burden on the latter. Responses are less likely to be biased or inconsistent and so less likely to lead to incorrect interpretations, in the same way as for the pairwise comparison methodology introduced in Section 2.5.2.

The procedure above is very important, as it yields a lexicon which balances the expressiveness–efficiency trade-off through recourse to human cognition. As such, we can expect it to be sufficient for describing the textures held in the Outex dataset introduced in Section 3.1, without being redundant or of too high a dimension. In the next section we use the pairwise comparison framework to label the Outex textures with the 98 attributes derived above.

3.3 Labelling task

The pairwise comparison framework introduces certain new challenges, however. For the Outex dataset of 3828 textures, there are $\binom{3828}{2} = \frac{3828 \times 3827}{2}$ comparisons possible for each of the 98 attributes: approximately 718 million comparisons in total. However, it has been indicated (Parikh and Grauman, 2011) that only relatively few comparisons – in the order of the number of items being compared – are needed to achieve results comparable to that of the complete case. We can also reduce the burden in the following ways:

- *Removing redundancy from the visual and semantic spaces*: Due to the PCA analysis of Bhushan et al. (1997) (see Section 3.2.2) this has already been performed to some extent for our semantic space, which was pared down to 98 texture words from an initial set of 367. Because our visual space of 3828 textures is actually composed of 319 textures captured at 12 different variations of illumination and rotation, we may easily reduce the dimensionality of this space by assuming comparisons to apply equally to all 12 samples within each texture class, owing to the natural human visual robustness to illumination and rotation when describing surface texture. As such, only textures with rotation of 0° and illumination of `horizon` need be displayed to users, and labels can be automatically propagated between the remaining 143 combinations of between-class samples. The number of unique pairwise comparisons for each attribute is thus $\binom{319}{2}$, for a total of around 5 million.
- *Exploiting similarities in the visual and semantic spaces*: Often many of the items in the visual space, and many attributes in the semantic space, are very similar

to other items in those spaces. When this is the case comparisons can be propagated so as to apply equally for similar items (and attributes), even if the original respondent did not view those items.

- *Selecting only the most informative comparisons*: Following from the previous point, the learning task can often be completed more quickly by always selecting the most informative comparison to show to respondents. In this way, the data elicitation task is made to be iterative, with the comparison shown to users at any point a function of all previous comparisons. Such a procedure is often known as *active learning.*

To address the latter two of these issues, we derive a new method for learning from pairwise comparisons informed by correlations in the visual and semantic spaces in Chapter 4. Also in that chapter, in Section 4.4.2, we quantitatively investigate the effect the number of comparisons has on predictive accuracy with a synthetic dataset.

Next, we describe the methodology employed for the labelling task.

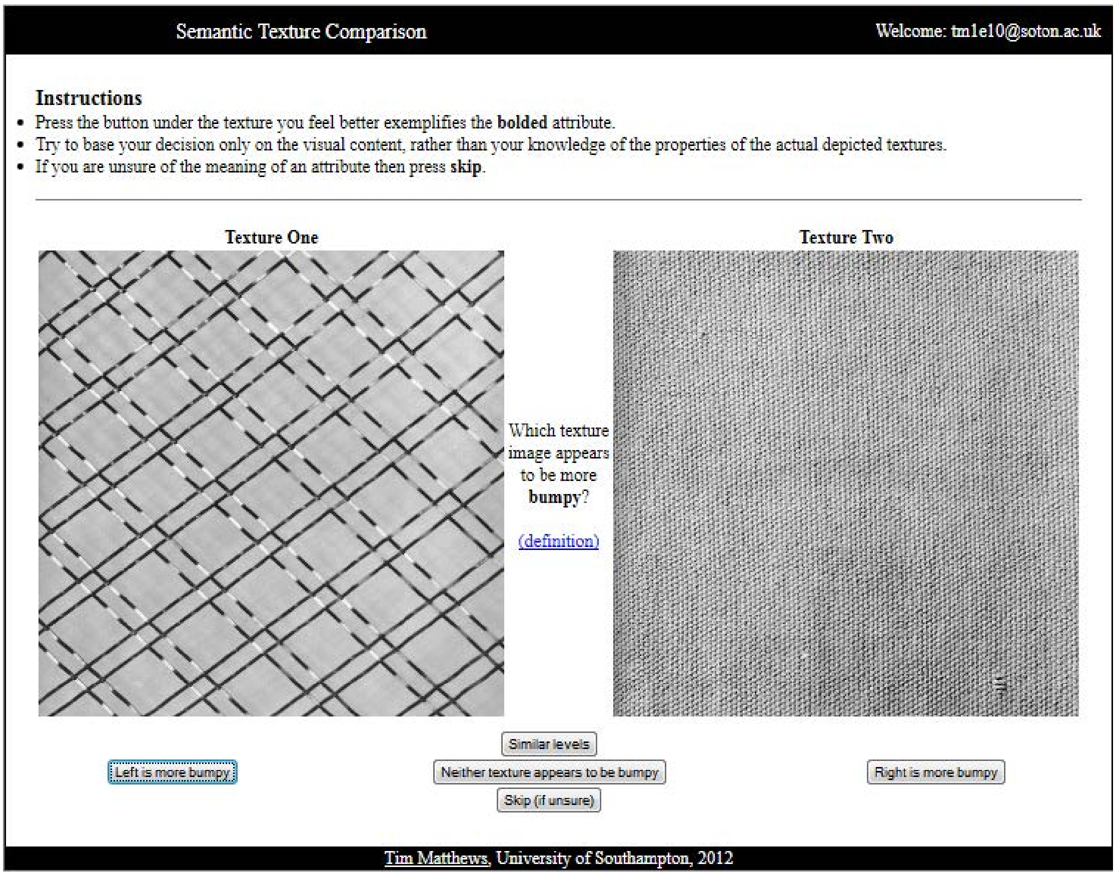
3.3.1 Methodology

Our methodology is as follows: a subject is shown two textures side-by-side along with a single attribute, as shown in Figure 3.3. The subject is prompted to select the texture that expresses a greater level of the attribute in question, or to rate them as similar.

To ensure the soundness of the basic procedure above we obtained a small number of comparisons in a limited pilot study conducted with around ten members of our research group. From this it was observed that pairs of textures appeared reasonably often which could not be said to be describable by the displayed attribute. Although computationally we would prefer respondents to mark pairs such as these as *similar* (as they both express similarly minimal levels of the attribute), it was observed that cognitively there was often a degree of hesitancy and doubt. To correct for this we added an extra option: "*Neither texture appears to be* `<adjective>`". Absence comparisons are equivalent to a similarity judgement in all subsequent analysis.

The attribute shown is the one involved in the fewest comparisons at that point. We made a deliberate design decision to target breadth rather than depth in the responses; that is, to achieve as many different pairwise comparisons as possible, rather than multiple responses from different subjects for each pairwise comparison used, so as to minimise the sparsity of the resultant dataset. To this end, the textures shown were chosen randomly among those that had not previously been directly compared.

Because of the massive amount of labels required, we used the Amazon Mechanical Turk (`https://www.mturk.com`) crowdsourcing platform to procure them, on which workers



Semantic Texture Comparison

Welcome: tm1e10@soton.ac.uk

Instructions

- Press the button under the texture you feel better exemplifies the **bolded** attribute.
- Try to base your decision only on the visual content, rather than your knowledge of the properties of the actual depicted textures.
- If you are unsure of the meaning of an attribute then press **skip**.

Texture One

Texture Two

Which texture image appears to be more **bumpy**?

(definition)

Tim Matthews, University of Southampton, 2012

FIGURE 3.3: Web interface used for obtaining texture comparisons.

are paid fees to perform small tasks for requesters. However, it is well-established that crowdsourced data can be unreliable (Kittur et al., 2008): respondents are anonymous, may be unfamiliar with the task at hand, and have a financial incentive to respond quickly, even at the expense of quality. To avoid the risk of garbage responses, potential workers were required to complete a brief English proficiency examination in order to submit labels. Each examination consisted of five multiple choice questions requiring answerers to select which of four words is most synonymous with another candidate word. Applicants required at least four correct answers out of the five in order to continue to the labelling task. In practice, the presence of the examination acted more as a deterrent to users with poor English proficiency: few people failed the examination, and almost all who took it scored five out of five. Later in this thesis, we will evaluate and prune the data from accepted workers before subsequent experimentation (Section 5.1).

Comparisons were sent to accepted workers in sets of ten, with $0.05 paid for a completed set. The labels obtained through Amazon Mechanical Turk are analysed in the next section.

3.3.2 Analysis

137,995 comparisons were obtained from 568 different respondents, averaging approximately 243 comparisons per respondent. Note that this is just 3% of the total possible $98 \times \binom{319}{2}$ comparisons. Table 3.2 shows respondents grouped by how many comparisons they made — around half of all comparisons were made by just the top 34 respondents, who each made 1000+ comparisons individually. This reliance on just a few dozen people meant our results were susceptible to noise from rogue users: we examine the dataset from a noise-reduction perspective in Section 5.1, identifying respondents whose overall contribution was deleterious to further analysis.

Comparisons	**Respondents**
1—10	155
11—100	202
101—1000	177
>1000	34
Total	568

Table 3.2: Groupings of respondents by how many comparisons they made.

Each attribute received just over 1400 comparisons on average, corresponding to around 9 comparisons per texture. The patterns of responses over all comparisons are displayed in Table 3.3. A large number of *neither* responses were received. Although these are still useful, as they map to similarity constraints in subsequent analysis, they suggest that the typical attribute only applied to a subset of the items, and some time could have been saved by inferring which item pairs were likely to be uninformative.

Response	**#**
Left	38,526
Right	40,612
Similar	7345
Neither	53,551
Total	137,995

Table 3.3: Number of comparisons for each response category.

To better illustrate our chosen attributes and how they relate to the Outex dataset we display in Appendix A an *exemplar* texture for each of the ninety-eight attributes, learned using the direct inference method of David (1987) introduced in Chapter 4.2.1. Empirically, we can see there is generally a good accordance between the exemplar texture and each attribute, giving an indication that any noise in our label dataset is outweighed by good responses. The results in the experiments detailed in Chapter 5 go on to support this suspicion quantitatively.

3.4 Summary

In this chapter we obtained a large crowdsourced dataset of pairwise comparison labels for a selection of texture images (Matthews, 2014). Although, we can measure the attribute strengths of the involved textures directly from the pairwise comparison data, it is desirable to be able to infer the semantic strength of our attributes for new, unseen textures. This can be done based only on their visual properties using the visual descriptors described previously. Methods for learning attributes strength ratings for both seen and unseen texture are described in the next chapter. We also derive a probabilistic method for inference from pairwise comparison data, which we eventually use to measure respondent reliability for the dataset collected above in Section 5.1.

Chapter 4

Bridging the Semantic Gap

Now that we have obtained a large dataset of labels, we may work on bridging the semantic gap by learning the correspondences between them so that we can automatically label, synthesise, and describe texture. To this end, in the current chapter we will outline current methods of learning from sets of pairwise comparisons; derive a new probabilistic method we may use to learn from sets of pairwise comparisons; and then evaluate the performance of this method on synthetic data.

We start by defining the problem of *bridging the semantic gap* more precisely in the next section.

4.1 Problem statement

In this section we introduce the notation used throughout this chapter and in the rest of the thesis, and formally state the problem to be solved – that of bridging the semantic gap.

4.1.1 General notation

We make use of various specialised operations on matrices.

The *Kronecker product* of two matrices, denoted $\mathbf{A} \otimes \mathbf{B}$ where $\mathbf{A}$ is $M \times N$ and $\mathbf{B}$ is $P \times Q$, is defined as the block matrix

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \cdots & a_{N,1}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \cdots & a_{N,2}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M,1}\mathbf{B} & a_{M,2}\mathbf{B} & \cdots & a_{M,N}\mathbf{B} \end{bmatrix} \tag{4.1}$$

and hence is of size $MP \times NQ$.

The *vectorised* form of a matrix, denoted $\mathrm{vec}(\mathbf{A})$, is a vector created by stacking each of the columns of its parameter matrix:

$$\mathrm{vec}(\mathbf{A}) = [a_{1,1}, \cdots, a_{M,1}, a_{1,2}, \cdots, a_{M,2}, \cdots, a_{1,N}, \cdots, a_{M,N}]^\top \tag{4.2}$$

Vectorisation is related to the Kronecker product through the equation below, which we make use of throughout this chapter:

$$\mathrm{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A})\mathrm{vec}(\mathbf{B}) \tag{4.3}$$

The *Moore-Penrose pseudoinverse* is a generalisation of the matrix inverse to non-square matrices (Penrose, 1956). The pseudoinverse of $\mathbf{A}$ is denoted $\mathbf{A}^\dagger$ and is a least squares solution to a system of linear equations:

$$\mathbf{A}^\dagger = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top \tag{4.4}$$

We now formally state the problem to be solved.

4.1.2 Bridging the semantic gap

We consider problems involving N items and P attributes. Each item $i = 1, \ldots, N$ is represented by a D-dimensional feature vector $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,D})$, occupying the i^{th} row of feature matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$. Each attribute $j = 1, \ldots, P$ is associated with an N-dimensional strength vector $\boldsymbol{r}_j = (r_{1,j}, \ldots, r_{N,j})^\top$, where $r_{i,j}$ is the unknown real-valued strength of attribute j for item i. Each $\boldsymbol{r}_j$ occupies the j^{th} column of an attribute strength matrix $\mathbf{R} \in \mathbb{R}^{N \times P}$, to be estimated as part of the learning problem.

Our response data is represented by a set $\mathcal{C}$ of M pairwise comparison tuples:

$$\mathcal{C} = \{(a, b, y)_1, (a, b, y)_2, \ldots, (a, b, y)_M\} \tag{4.5}$$

where for tuple $(a, b, y)_i$, $a, b \in \{1, \ldots, N\}$, $y \in \{-1, 0, 1\}$ and $a \neq b$. $y = 1$ indicates the respondent felt that item a exhibited a greater quantity of attribute p than item b; $y = -1$ indicates the opposite; and $y = 0$ indicates the respondent felt both a and b exhibited similar levels of the attribute.

From this we define the sets of dominance relations, $\mathcal{C}^{\mathcal{D}}$, and similarity relations, $\mathcal{C}^{\mathcal{S}}$, as:

$$\mathcal{C}^{\mathcal{D}} = \{(a, b, y)_i | (a, b, y)_i \in \mathcal{C} \wedge y \in \{-1, 1\}\} \tag{4.6}$$

$$\mathcal{C}^{\mathcal{S}} = \{(a, b, y)_i | (a, b, y)_i \in \mathcal{C} \wedge y = 0\} \tag{4.7}$$

Similarly, for the p^{th} attribute, we obtain a total of $M^{(p)}$ pairwise comparison tuples:

$$\mathcal{C}^{(p)} = \left\{(a,b,y)_1^{(p)}, (a,b,y)_2^{(p)}, \ldots, (a,b,y)_{M^{(p)}}^{(p)}\right\}$$

For mathematical convenience, we encode these tuples in a sparse matrix, $\mathbf{C}^{(p)} \in \{-1,0,1\}^{M^{(p)} \times N}$ and a response vector $\boldsymbol{y}^{(p)} \in \{-1,0,1\}^{M^{(p)} \times 1}$, where for tuple $(a,b,y)_i^{(p)}$, $\mathbf{C}_{i,a}^{(p)} = 1$, $\mathbf{C}_{i,b}^{(p)} = -1$, and $\boldsymbol{y}_i^{(p)} = y$. We also define the (unknown) rating difference for each pairwise comparison:

$$\boldsymbol{z}^{(p)} = \mathbf{C}^{(p)} \boldsymbol{r}_p \tag{4.8}$$

such that $z_i^{(p)} = r_{a,p} - r_{b,p}$

Next, we introduce structures that contain data concerning pairwise comparisons across *all* attributes. In this way: $M = \sum_i^P M^{(i)}$, and:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}^{(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^{(2)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}^{(P)} \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}^{(1)} \\ \boldsymbol{y}^{(2)} \\ \vdots \\ \boldsymbol{y}^{(P)} \end{bmatrix} \qquad \boldsymbol{z} = \begin{bmatrix} \boldsymbol{z}^{(1)} \\ \boldsymbol{z}^{(2)} \\ \vdots \\ \boldsymbol{z}^{(P)} \end{bmatrix} \tag{4.9}$$

Hence, $\mathbf{C} \in \{-1,0,1\}^{M \times NP}$, $\boldsymbol{y} \in \{-1,0,1\}^{M \times 1}$ and $\boldsymbol{z} \in \mathbb{R}^{M \times 1}$. Following from Equation 4.8, $\boldsymbol{z}$ is calculable directly from $\mathbf{C}$ and $\mathbf{R}$ as:

$$\boldsymbol{z} = \mathbf{C}\text{vec}(\mathbf{R}) \tag{4.10}$$

Given a set of pairwise comparisons held in $\mathbf{C}$ and $\boldsymbol{y}$ it is of great interest to be able to accurately estimate the unknown attribute strength matrix $\mathbf{R}$. However, our goal in bridging the semantic gap is to be able to assess the attribute strength ratings for *unseen* textures and attributes which weren't involved in the initial labelling process, a problem known as *learning-to-rank.* For this reason any learning mechanism suitable for bridging the semantic gap should not only accurately estimate $\mathbf{R}$, but also provide some means for generalisation.

In the next section we outline selected techniques for learning item ratings from pairwise comparisons, both in the case where the ratings are learned directly from the comparisons, and in the case where the ratings are calculated by applying a learned ranking function to the item features.

4.2 Learning rankings

In this section we introduce two fundamentally different methods of estimating attribute strength ratings from pairwise comparisons. The first is a statistical technique operating directly on the pairwise comparison graph and hence cannot be used to infer ratings for unseen items, but is useful in cases where generalisation is not required and a feature space has not been defined. The second – RankSVM – belongs to a class of inference algorithms known as *learning-to-rank* where generalisation is possible through learning a ranking function which maps feature space representations of items to a real-valued ranking value. We start the section with an example of the former of these.

4.2.1 Statistical inference

Often it is useful to be able to directly infer attribute strengths from pairwise comparison data, even when a feature space representation is unavailable or undesirable. Doing so allows us to answer simple questions about the dataset – as we did when calculating the exemplar textures in Appendix A – and may serve as a ground truth measure against which the outputs ranking functions can be judged. Such rankings could also be used to derive semantic modifiers to our basic attribute forms, allowing constructs such as "*slightly* spiralled" or "*extremely* bumpy".

Inference methods of this kind often proceed using a *preference* matrix, $\mathbf{A}$, where $\mathbf{A}_{ij}$ is the number of times item i was preferred over item j, with similarity relations counting as half a dominance relation (Andrews and David, 1990). Item win and loss totals are calculated from the row and column sum respectively:

$$\begin{aligned} \boldsymbol{w} &= \mathbf{A}\mathbf{1}_N \\ \boldsymbol{l} &= \mathbf{A}^\top\mathbf{1}_N \end{aligned} \tag{4.11}$$

where $\boldsymbol{w}$ is a vector of item win totals, $\boldsymbol{l}$ is a vector of item loss totals, and $\mathbf{1}_N$ is an N-length vector of ones.

In this thesis we use the simple mechanism introduced by David (1987) designed to operate on incomplete sets of pairwise comparisons such as ours. In this system the strength ratings for an attribute, $\boldsymbol{r}$, are calculated as:

$$\boldsymbol{r} = \boldsymbol{w} - \boldsymbol{l} + \mathbf{A}\boldsymbol{w} - \mathbf{A}^\top\boldsymbol{l} \tag{4.12}$$

Intuitively, the method works by scoring item i as follows:

- The score of item i is the number of times i dominates another item…
- …plus the number of times *those* items dominate any other item…

- . . . minus the number of times i is dominated by another item. . .
- . . . minus the number of times *those* items are dominated by any other item.

Using the notation that r_i is the score of item i and that $\mathcal{C}^{\mathcal{D}}$ is a set of dominance orderings as defined in Equation 4.6, we can present the above scoring system as follows:

$$r_i = \sum_{(i,a)\in\mathcal{C}^{\mathcal{D}}} (1 + \sum_{(a,b)\in\mathcal{C}^{\mathcal{D}}} 1) - \sum_{(a,i)\in\mathcal{C}^{\mathcal{D}}} (1 + \sum_{(b,a)\in\mathcal{C}^{\mathcal{D}}} 1) \tag{4.13}$$

Similarity orderings, held in $\mathcal{C}^{\mathcal{S}}$, are introduced by considering similar items as having half the effect of dominating and half the effect of being dominated.

$$w_i = \sum_{(i,j)\in\mathcal{C}^{\mathcal{D}}} 1 + \sum_{(i,j)\in\mathcal{C}^{\mathcal{S}}} \frac{1}{2} \tag{4.14}$$

$$l_i = \sum_{(j,i)\in\mathcal{C}^{\mathcal{D}}} 1 + \sum_{(i,j)\in\mathcal{C}^{\mathcal{S}}} \frac{1}{2} \tag{4.15}$$

$$r_i = w_i - l_i + \sum_{(i,j)\in\mathcal{C}^{\mathcal{D}}} w_j - \sum_{(j,i)\in\mathcal{C}^{\mathcal{D}}} l_j + \sum_{(i,j)\in\mathcal{C}^{\mathcal{S}}} \frac{1}{2}(w_j - l_j) \tag{4.16}$$

A ranking can then be derived naturally from these scores by sorting in descending order. In the next section we introduce the Ranking SVM – a technique for learning a ranking function from a feature space representation of the items, such that items uninvolved in the pairwise comparison procedure can also be rated.

4.2.2 Ranking SVM

The traditional Support Vector Machine (SVM) aims to determine the optimally separating hyperplane between the feature space representations of two classes of data. Using $\boldsymbol{x}_i$ to represent the location in feature space of the i^{th} item in a dataset of N items, y_i to denote the classification (either 1 or -1) of that item, and $\boldsymbol{w}$ as the normal vector to the separating hyperplane, we can display this aim in the form of an optimisation problem:

$$\begin{aligned} &\underset{\boldsymbol{w}}{\text{minimise}} && \frac{1}{2}||\boldsymbol{w}||^2 \\ &\text{subject to} && y_i \boldsymbol{w} \cdot \boldsymbol{x}_i \geq 1,\ i = 1, \ldots, N \end{aligned} \tag{4.17}$$

In cases where the two classes of data are not linearly separable within the feature space a soft-margin SVM (Cortes and Vapnik, 1995) can be used. This formulation introduces so-called *slack variables*, where ξ_i is the misclassification error of item i. The quadratic sum of the slack variables is included as a term in the objective function, with C acting as the trade-off between maximising the margin and minimising the misclassification error:

$$\begin{aligned} \underset{\boldsymbol{w}}{\text{minimise}} \quad & \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{N}\xi_i^2 \\ \text{subject to} \quad & y_i\boldsymbol{w}\cdot\boldsymbol{x}_i \geq 1-\xi_i,\ \xi_i \geq 0,\ i=1,\ldots,N \end{aligned} \tag{4.18}$$

The Ranking SVM (Joachims, 2002) is an analog to the traditional SVM with the aim of determining the optimally separating hyperplane between *differences* of feature vectors, rather than between the feature vectors themselves. These difference vectors are classified according to whether they represent a *greater-than* or *less-than* relation.

The Ranking SVM without similarity constraints is thus formulated as:

$$\begin{aligned} \underset{\boldsymbol{w}}{\text{minimise}} \quad & \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{M}\xi_i^2 \\ \text{subject to} \quad & y\boldsymbol{w}\cdot(\boldsymbol{x}_a-\boldsymbol{x}_b) \geq 1-\xi_i,\ \xi_i \geq 0,\ (a,b,y)_i \in \mathcal{C}^{\mathcal{D}} \end{aligned} \tag{4.19}$$

In this context $\boldsymbol{w}$ is interpreted not just as a separating hyperplane between *greater-than* and *less-than* relations, but also as a *ranking function* capable of mapping new, unseen feature vectors to a real-valued measure of attribute strength.

Support for similarity constraints is gained using the formulation of Parikh and Grauman (2011):

$$\begin{aligned} \underset{\boldsymbol{w}}{\text{minimise}} \quad & \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{M}\xi_i^2 \\ \text{subject to} \quad & y\boldsymbol{w}\cdot(\boldsymbol{x}_a-\boldsymbol{x}_b) \geq 1-\xi_i,\ (a,b,y)_i \in \mathcal{C}^{\mathcal{D}} \\ & \boldsymbol{w}\cdot(|\boldsymbol{x}_a-\boldsymbol{x}_b|) \leq \xi_i, (a,b,y)_i \in \mathcal{C}^{\mathcal{S}} \\ & \xi_i \geq 0 \end{aligned} \tag{4.20}$$

This formulation departs slightly from that of the traditional SVM structure and so cannot be solved using specialised solvers. However, traditional quadratic programming approaches still apply.

4.2.3 Summary

In light of the introductions given above, it is helpful to formulate a list of properties a system for inferring attribute ratings might possess, paying no regard at this point to whether these properties are jointly achievable:

- *Accuracy* — To deliver a ratings estimate $\hat{\mathbf{R}}$ that reflects as accurately as possible the true ratings $\mathbf{R}$ given a set of pairwise comparisons $\mathcal{C}$.
- *Resilience* — To be able to deliver this estimate even if $\mathcal{C}$ is incomplete – it does not hold every single possible comparison – or includes erroneous labels.

- *Time efficiency* — To be able to deliver the estimate as efficiently as possible as the dimensionality of $\mathcal{C}$ increases.
- *Certainty* — To quantify the degree of uncertainty in its estimates.
- *Generalisability* — To be able to generalise and deliver rating estimates for new, unseen items and attributes. It is this property that differentiates a learning-to-rank approach from a purely statistical one.

The Ranking SVM possesses many of these properties, and is hence one of the most widely used learning-to-rank techniques for pairwise comparison datasets: it is *accurate*, and is in a sense optimal in that it will obtain the maximally separating hyperplane for a set of comparisons; this optimisation problem can be solved in a *time efficient* manner; it is somewhat *resilient* due to the regularisation term $||\boldsymbol{w}||^2$ in the objective function protecting against overfitting; and it outputs a ranking function allowing it to easily *generalise* to new, unseen items.

However, we argue that it is deficient in situations where there are only relatively few pairwise comparisons. In these situations it may be expensive to obtain the requisite number of comparisons to properly rate a set of items with confidence, and information about the structures of the item and attribute spaces could be used to better infer missing comparisons. In fact, the Ranking SVM does not incorporate any information about the attribute space at all: each attribute must be learned individually and it is not possible to generalise the output to new attributes. Furthermore, the *degree* of certainty is unknown, and only point estimates of item ratings are provided.

In the next section we derive a Bayesian probabilistic approach that addresses these concerns by learning a probability distribution over the sought after ratings based on information about the structure of the item *and* attribute spaces.

4.3 Probabilistic inference from pairwise comparisons

This section derives a probability distribution over the unknown attribute ratings, $\Pr(\mathbf{R}|\mathcal{C})$. As we intend for the framework to operate with large numbers of attributes whose visual form may be unknown or difficult to capture *a priori*, we wish to avoid using any explicit feature space representation for inference. This is unlike traditional learning-to-rank formulations, which take a feature matrix, $\mathbf{X}$, to have been fully defined.

Instead, due to the nature of our pairwise comparison dataset, we choose to make use of similarity measures between both textures and attributes. Loosely, if we are sure a particular texture is visually similar to some other texture, and an attribute has similar meaning to some other attribute, then it is reasonable to assume both textures express both attributes at similar levels, even if we have no other feature information about the textures or attributes. This is a particularly natural framing for our attributes, them being simple atomic descriptive units for which a feature space representation is not forthcoming in the same way as for say, textures, which are backed by raw signal data.

To this end, we assume we instead provide a symmetric matrix of between-item similarities $\mathbf{S} \in \mathbb{R}^{N \times N}$, and between-attribute similarities, $\mathbf{T} \in \mathbb{R}^{P \times P}$. To be precise, in our probabilistic formulation these similarity matrices take the form of covariance (or correlation) matrices, and can be defined in numerous ways depending on the problem at hand and the nature of the space. For instance, similarities could be calculated using a computational distance measure based on some underlying feature space representation; by a psychometric procedure such as that performed by Bhushan et al. (1997) (and outlined in Section 3.2.2); using specialised notions of similarity such as those developed for semantic distance in WordNet(Budanitsky and Hirst, 2001); or handcrafted using expert knowledge of the structure.

In this way $\mathbf{S}$ and $\mathbf{T}$ effectively act as an implicit feature space without requiring an explicit feature selection stage, and they introduce an appealing symmetry between items and attributes: both spaces have identical mathematical structures and are given equal prominence within the formulation, in contrast to traditional "item-centric" approaches. This symmetry permits generalisation across both items *and* attributes, with ratings for new, unseen categories learned by evaluating their similarity to existing items in an appropriate way, updating $\mathbf{S}$ and $\mathbf{T}$, and then re-calculating $\Pr(\mathbf{R}|\mathcal{C}, \mathbf{S}, \mathbf{T})$. We shall also see later in this chapter that this transposability allows for an entire set of ratings across multiple items and attributes to be jointly learned, rather than inferred one attribute at a time.

We begin the derivation of the probabilistic framework in the next section.

4.3.1 Multivariate normal distribution

The multivariate normal distribution is a multivariate generalization of the normal distribution. It allows us to impose a probabilistic structure on a K-dimensional vector. Parametrised with a mean vector, $\boldsymbol{\mu} \in \mathbb{R}^{K \times 1}$, and covariance matrix, $\boldsymbol{\Sigma} \in \mathbb{R}^{K \times K}$, it is defined as:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi^{\frac{K}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}})^{-1} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{4.21}$$

As the support of the multivariate normal distribution is a vector in $\mathbb{R}^{K \times 1}$, we can seek to learn the parameters for this distribution over the unknown vector of strength ratings for a single attribute, denoted $\boldsymbol{r}_p$ for attribute p, and hence quantify the uncertainty in these ratings in a principled way.

To do this, we recall notation from Section 4.1.2 differentiating between the *actual* (unknown) strength differences, denoted $\boldsymbol{z}^{(p)}$, and the *perceived* (subjective) strength difference response, which are instead denoted $\boldsymbol{y}^{(p)}$. Although we have defined $\boldsymbol{y}^{(p)}$ to take values in the set $\{-1, 0, 1\}$ corresponding to "*less than*", "*similar to*", and "*greater than*" judgements respectively, for now we will proceed as if responses in $\boldsymbol{y}^{(p)}$ can take any real value, such that we can interpret them probabilistically as being randomly perturbed instances of the unknown strength differences. This perturbation can account for subjective variations in how the attributes and visual textures are interpreted as well as accounting for noise and other effects. We represent these perturbations in a random error vector, $\boldsymbol{\epsilon}$, like so:

$$\boldsymbol{y}^{(p)} = \boldsymbol{z}^{(p)} + \boldsymbol{\epsilon} \tag{4.22}$$

However, as mentioned, for the dataset we collected in the previous chapter the exact nature of this perturbation is complicated by the fact we don't have available a real-valued measure of each response (although a different experimental setup could provide this). Instead, for experimental simplicity we elicited only relative judgements reflecting dominance or similarity. Distributions involving truncation and discontinuities such as this are typically less convenient to work with. Although in principle the resulting treatment could proceed using any noise model the eventual Bayesian analysis is made especially simple when using distributions with certain properties. In particular, the normal (and multivariate normal) distributions are particularly simple to work with analytically, particularly when it comes to an eventual Bayesian formulation, and throughout this section we proceed under the assumption of normally distributed noise. Gaussian noise of this kind is a key assumption of our model. Furthermore we take $\boldsymbol{\epsilon}$ to be *homoscedastic* – each error term has the same variance, equal to σ^2. However, if more information were known about the biases and tendencies of the respondents then the covariance matrix could be formulated to better reflect the subjective error. For simplicity, we proceed using homoscedastic Gaussian noise.

With the above in mind, by assuming each element of the error vector is independent and normally distributed with zero mean and variance σ^2, we have:

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I}) \tag{4.23}$$

Combining equations Equation 4.22 and Equation 4.23 yields a probabilistic form for $\boldsymbol{y}^{(p)}$, the *perceived* strength differences:

$$\boldsymbol{y}^{(p)} \sim \mathcal{N}(\boldsymbol{z}^{(p)}, \sigma^2\mathbf{I}) \tag{4.24}$$

In Equation 4.8 we have defined $\boldsymbol{z}^{(p)}$ such that it can be obtained by multiplying our comparison matrix $\mathbf{C}^{(p)}$ by the vector of unknown strength ratings $\boldsymbol{r}_p$. As such, we can substitute $\boldsymbol{z}$ with $\mathbf{C}\boldsymbol{r}$ to obtain:

$$\boldsymbol{y} \sim \mathcal{N}(\mathbf{C}\boldsymbol{r}, \sigma^2\mathbf{I}) \tag{4.25}$$

Here and from this point on we will stop using the attribute-specific $^{(p)}$ indicator when discussing derivations based on the multivariate normal distribution, when it is clear through context that only a single attribute is being discussed.

The probability of a subject perceiving the rating differences held in $\boldsymbol{y}$ is therefore:

$$\begin{aligned}
\Pr(\boldsymbol{y}|\mathbf{C}, \boldsymbol{r}, \sigma^2) &= \mathcal{N}(\boldsymbol{y}|\mathbf{C}\boldsymbol{r}, \sigma^2\mathbf{I}) && (4.26)\\
&= (\frac{1}{2\pi\sigma^2})^{M/2} \exp\left(-\frac{1}{2\sigma^2}\boldsymbol{\delta}^\top\boldsymbol{\delta}\right) && (4.27)\\
\boldsymbol{\delta} &= \boldsymbol{y} - \mathbf{C}\boldsymbol{r} && (4.28)
\end{aligned}$$

Our goal is to infer the unknown attribute strengths $\boldsymbol{r}$ from observations of attribute strength differences, $\boldsymbol{y}$. In the next section we do this using maximum likelihood and later using a Bayesian inference scheme.

4.3.1.1 Maximum likelihood estimate

The maximum likelihood estimate of the mean parameter of this distribution is given by the sample mean of the data. The log-likelihood function is:

$$\begin{aligned}
\ln \mathcal{L}(\boldsymbol{r}, \sigma^2|\boldsymbol{y}) &= \ln \Pr(\boldsymbol{y}|\mathbf{C}, \boldsymbol{r}, \sigma^2) && (4.29)\\
&= -\frac{M}{2}\ln 2\pi - \frac{M}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\boldsymbol{\delta}^\top\boldsymbol{\delta} && (4.30)\\
\boldsymbol{\delta} &= \boldsymbol{y} - \mathbf{C}\boldsymbol{r} && (4.31)
\end{aligned}$$

Differentiating with respect to the attribute ratings gives:

$$\frac{\partial \ln \mathcal{L}}{\partial \boldsymbol{r}} = -\frac{1}{2\sigma^2}\left(-2\mathbf{C}^\top\boldsymbol{y} + 2\mathbf{C}^\top\mathbf{C}\boldsymbol{r}\right) \tag{4.32}$$

Setting this equal to zero and solving yields:

$$\begin{aligned}
\hat{\boldsymbol{r}}^{\mathrm{MLE}} &= (\mathbf{C}^\top\mathbf{C})^{-1}\mathbf{C}^\top\boldsymbol{y} && (4.33)\\
&= \mathbf{C}^\dagger\boldsymbol{y} && (4.34)
\end{aligned}$$

This demonstrates the well-known result that the maximum likelihood estimate in this setting is equivalent to the matrix solution to the system of linear equations given by Equation 4.8.

However, this maximum likelihood treatment is limited in many ways. It provides just a single fixed estimate of $\boldsymbol{r}$, and we have no means of quantifying the uncertainty associated with this estimate. It is a common trait of maximum likelihood estimators that they are prone to overfitting, and the sparseness of our crowdsourced dataset makes this a particularly pertinent issue, exacerbated by the fact we have no means of imposing additional structure upon the estimated ratings. We wish to use our knowledge of the texture and semantic spaces to influence the output of the inference, and to then be able to measure the uncertainty associated with that output. These goals are possible with *Bayesian* techniques, which we introduce in the next section.

4.3.1.2 Bayesian inference

In a Bayesian approach to inference our *prior* belief of the distribution some unknown parameters take, $\Pr(\Omega)$, is updated upon observation of some data, $\mathcal{X}$, so as to obtain a new *posterior* belief conditional on this data:

$$\Pr(\Omega|\mathcal{X}) = \frac{\Pr(\mathcal{X}|\Omega)\Pr(\Omega)}{\Pr(\mathcal{X})} \tag{4.35}$$

This is *Bayes' rule*. Importantly, this rule allows us to quantify our uncertainty over the parameters through inspection of the posterior, and to avoid overfitting the parameters to the data through incorporation of prior knowledge of the parameter structure. For particular forms of the likelihood distribution, $\Pr(\mathcal{X}|\Omega)$, it is possible to choose a prior distribution which will yield a posterior distribution of the same analytical form — such priors are known as *conjugate priors*.

For the problem at hand, the data likelihood function Equation 4.26 is a multivariate normal distribution. Here, the mean is governed by the unknown attribute ratings, $\boldsymbol{r}$, and the covariance is scaled by σ^2. The normal-inverse-gamma distribution is a conjugate prior over the mean and variance of a multivariate normal distribution (Gelman et al., 1995), and so:

$$\begin{aligned}
\Pr(\boldsymbol{r},\sigma^2) &= \mathcal{NG}^{-1}(\boldsymbol{r},\sigma^2|\boldsymbol{m},\boldsymbol{\Sigma},a,b) && (4.36)\\
&= \mathcal{N}(\boldsymbol{r}|\boldsymbol{m},\sigma^2\boldsymbol{\Sigma})\mathcal{G}^{-1}(\sigma^2|a,b) && (4.37)\\
&= \frac{b^a}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}\Gamma(a)}\Big(\frac{1}{\sigma^2}\Big)^{a+\frac{D}{2}+1}\exp\Big(-\frac{1}{\sigma^2}\Big[b+\frac{1}{2}\boldsymbol{\delta}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\delta}\Big]\Big) && (4.38)\\
\boldsymbol{\delta} &= \boldsymbol{r}-\boldsymbol{m} && (4.39)
\end{aligned}$$

where $\boldsymbol{m}$ and $\boldsymbol{\Sigma}$ are the prior normal mean and covariance hyperparameters, and a and b are hyperparameters for the inverse gamma distribution over σ^2. By setting these hyperparameters appropriately, we are able to impose additional structure on the eventual estimate of $\boldsymbol{r}$. In particular, the between-item similarity matrix $\mathbf{S}$ described in the introduction to Section 4.3 can be used to inform the prior covariance, $\mathbf{\Sigma} = \mathbf{S}$, in turn influencing $\boldsymbol{r}$.

From Bayes' rule (Equation 4.35), the posterior distribution is calculated as proportional to the product of Equation 4.36 (the prior) and Equation 4.26 (the likelihood). Because the normal-inverse-gamma is a conjugate prior for a multivariate normal likelihood distribution with unknown mean and variance, the posterior is also a normal-inverse-gamma:

$$\begin{aligned}\Pr(\boldsymbol{r}, \sigma^2 | \boldsymbol{y}) &\propto \mathcal{N}(\boldsymbol{y} | \boldsymbol{r}, \sigma^2 \mathbf{I}) \mathcal{NG}^{-1}(\boldsymbol{r}, \sigma^2 | \boldsymbol{m}, \mathbf{\Sigma}, a, b) && (4.40)\\ &= \mathcal{NG}^{-1}(\boldsymbol{r}, \sigma^2 | \boldsymbol{m}_*, \mathbf{\Sigma}_*, a_*, b_*) && (4.41)\end{aligned}$$

where:

$$\begin{aligned}\mathbf{\Sigma}_* &= (\mathbf{\Sigma}^{-1} + \mathbf{C}^\top \mathbf{C})^{-1} && (4.42)\\ \boldsymbol{m}_* &= \mathbf{\Sigma}_* (\mathbf{\Sigma}^{-1} \boldsymbol{m} + \mathbf{C}^\top \boldsymbol{y}) && (4.43)\\ a_* &= a + \frac{M}{2} && (4.44)\\ b_* &= b + \frac{1}{2} (\boldsymbol{m}^\top \mathbf{\Sigma}^{-1} \boldsymbol{m} + \boldsymbol{y}^\top \boldsymbol{y} - \boldsymbol{m}_*^\top \mathbf{\Sigma}_*^{-1} \boldsymbol{m}_*) && (4.45)\end{aligned}$$

Next, we marginalise over σ^2 to obtain a distribution over $\boldsymbol{r}$. This is a characteristic feature of Bayesian methods compared to frequentist approaches such as maximum likelihood: we are able to eliminate so-called *nuisance parameters* such as σ^2 to obtain predictive distributions purely over the parameters of interest $\boldsymbol{r}$ which incorporate all our uncertainty surrounding σ^2. The marginal distribution of the mean vector of a normal-inverse-gamma distribution is a multivariate t-distribution (Gelman et al., 1995):

$$\begin{aligned}\Pr(\boldsymbol{r} | \boldsymbol{y}) &= \int_{-\infty}^{\infty} \Pr(\boldsymbol{r}, \sigma^2 | \boldsymbol{y}) \, \mathrm{d}\sigma && (4.46)\\ &= \int_{-\infty}^{\infty} \mathcal{NG}^{-1}(\boldsymbol{r}, \sigma^2 | \boldsymbol{m}_*, \mathbf{\Sigma}_*, a_*, b_*) \, \mathrm{d}\sigma && (4.47)\\ &= \mathcal{T}(\boldsymbol{r} | \boldsymbol{m}_*, \frac{b_*}{a_*} \mathbf{\Sigma}_*, 2a_*) && (4.48)\\ &= \frac{\Gamma(a_* + \frac{N}{2})}{\Gamma(a_*) \pi^{\frac{N}{2}} |2b_* \mathbf{\Sigma}_*|^{\frac{1}{2}}} \left(1 + \frac{1}{2b_*} \boldsymbol{\delta}^\top \mathbf{\Sigma}_*^{-1} \boldsymbol{\delta}\right)^{-a_* - \frac{N}{2}} && (4.49)\\ \boldsymbol{\delta} &= \boldsymbol{r} - \boldsymbol{m}_* && (4.50)\end{aligned}$$

The primary output of Bayesian inference is the posterior distribution over the parameters, and often subsequent analysis can operate directly on this complete distribution. Occasionally it is desirable to obtain a point estimate for the parameters. This is typically done by identifying the mode of the posterior distribution, as this is the parameter value for which the posterior probability is highest. This is known as a *maximum a posteriori* (MAP) estimate.

As the mode of a multivariate t-distribution is $\boldsymbol{m}_*$, the MAP estimate for $\boldsymbol{r}$ is:

$$\hat{\boldsymbol{r}}^{\mathrm{MAP}} = (\boldsymbol{\Sigma}^{-1} + \mathbf{C}^\top \mathbf{C})^{-1}(\boldsymbol{\Sigma}^{-1}\boldsymbol{m} + \mathbf{C}^\top \boldsymbol{y}) \tag{4.51}$$

Likewise, the MAP estimate of the error variance, σ^2, is the mode of an inverse Gamma distribution, and equal to:

$$\hat{\sigma^2}^{\mathrm{MAP}} = \frac{b_*}{a_* + 1} \tag{4.52}$$

Although not of direct interest here, we demonstrate in Section 5.1 how $\hat{\sigma^2}^{\mathrm{MAP}}$ can be used to identify respondents who provided poor quality labels.

Also worth noting is the posterior predictive distribution over the responses $\boldsymbol{y}^\star$ to newly observed comparisons held in $\mathbf{C}^\star$:

$$\Pr(\boldsymbol{y}^\star|\boldsymbol{y}) = \int \Pr(\boldsymbol{y}^\star|\boldsymbol{r}_p, \sigma^2) \Pr(\boldsymbol{r}_p, \sigma^2|\boldsymbol{y}) \,\mathrm{d}\boldsymbol{r}_p \,\mathrm{d}\sigma^2 \tag{4.53}$$

$$= \int \mathcal{N}(\boldsymbol{y}^\star|\mathbf{C}^\star\boldsymbol{r}_p, \sigma^2\mathbf{I})\mathcal{NG}^{-1}(\boldsymbol{r}_p, \sigma^2|\boldsymbol{m}_*, \boldsymbol{\Sigma}_*, a_*, b_*) \,\mathrm{d}\boldsymbol{r}_p \,\mathrm{d}\sigma^2 \tag{4.54}$$

$$= \mathcal{T}(\boldsymbol{y}^\star|\mathbf{C}^\star\boldsymbol{m}_*, \frac{b_*}{a_*}(\mathbf{C}^\star\boldsymbol{\Sigma}_*\mathbf{C}^{\star\top} + \mathbf{I}), 2a_*) \tag{4.55}$$

The posterior predictive distribution gives the most complete account of the uncertainty surrounding the responses in a new dataset, taking into consideration both the model uncertainty captured by σ^2, as well as the uncertainty in the posterior distribution due to the data. It is well-suited to areas involving human-computer interaction such as *active learning* (Settles, 2009), where an algorithm designed to learn from some dataset plays an active part in the dataset's construction by presenting training examples to users that will most improve its performance. An active learning variety of our labelling task described in Section 3.3 might use the posterior predictive distribution to determine the texture pair with the greatest response uncertainty, present this pair to a respondent for labelling, and then update its posterior and repeat for the next label.

The Bayesian approach derived above gives us a full distribution over $\boldsymbol{r}$ rather than just a point estimate, and — through Σ, $\boldsymbol{m}$, a, and b — allows us to incorporate our prior knowledge of the visual space. However as it is based on a multivariate normal distribution over *vector* quantities, we must still estimate the strengths for each attribute independently, depriving ourselves of the ability to better inform our ratings using the between-attribute similarities in $\mathbf{T}$. In the next section we outline an approach to jointly estimating $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_p$ using a distribution over *matrix* quantities.

4.3.2 Matrix-variate normal distribution

The matrix-variate normal distribution is a generalization of the multivariate normal distribution with support over matrices rather than vectors. It has one matrix-valued location parameter ($\mathbf{M} \in \mathbb{R}^{K\times P}$) and two covariance matrices ($\mathbf{U} \in \mathbb{R}^{K\times K}, \mathbf{V} \in \mathbb{R}^{P\times P}$) governing row and column covariance respectively. Its probability density function is:

$$\mathcal{N}(\mathbf{A}|\mathbf{M},\mathbf{U},\mathbf{V}) = (2\pi^{\frac{KP}{2}}|\mathbf{U}|^{\frac{P}{2}}|\mathbf{V}|^{\frac{K}{2}})^{-1}\exp\left(-\frac{1}{2}\mathrm{tr}\big[\mathbf{V}^{-1}(\mathbf{A}-\mathbf{M})^\top\mathbf{U}^{-1}(\mathbf{A}-\mathbf{M})\big]\right) \tag{4.56}$$

It is well-suited to certain *transposable* datasets in which both the rows and columns have equal claim as features, such as in recommender systems where it has been used to model between-viewer and between-film similarities for the Netflix prize (Allen and Tibshirani, 2010).

Every matrix-variate normal distribution can be represented by an equivalent multivariate normal distribution (Gupta and Nagar, 1999):

$$\mathcal{N}(\mathbf{A}|\mathbf{M},\mathbf{U},\mathbf{V}) = \mathcal{N}(\mathrm{vec}(\mathbf{A})|\mathrm{vec}(\mathbf{M}),\mathbf{V}\otimes\mathbf{U}) \tag{4.57}$$

It is appealing to harness the matrix-variate normal distribution to fully model the between-item and between-attribute structure belying the attribute strengths, $\mathbf{R}$. For the multivariate normal case described in the previous section—and recalling the attribute-specific $^{(i)}$ notation described in Section 4.1.2—the complete probability distribution over the perceived rating differences $\boldsymbol{y}$ is:

$$\Pr(\boldsymbol{y}|\mathbf{C},\mathbf{R}) = \prod_{i=1}^{P}\mathcal{N}(\boldsymbol{y}^{(i)}|\mathbf{C}^{(i)}\boldsymbol{r}_i,\sigma^2\mathbf{I}) \tag{4.58}$$

By using the vectorisation of $\mathbf{R}$ and Equation 4.57, we can express this equivalently without the product as:

$$\Pr(\boldsymbol{y}|\mathbf{C},\mathbf{R}) = \mathcal{N}(\boldsymbol{y}|\mathbf{C}\mathrm{vec}(\mathbf{R}),\sigma^2\mathbf{I}) \tag{4.59}$$

Note that this continues to treat each individual comparison response as an independent observation corrupted by normally-distributed errors with variance σ^2, having an identical form as in the single attribute case shown in Equation 4.25. We can thus proceed in an identical fashion as previously by imposing a normal-inverse-gamma prior distribution over $\mathbf{R}$ and σ^2:

$$\begin{aligned}\Pr(\mathbf{R},\sigma^2) &= \mathcal{NG}^{-1}(\mathrm{vec}(\mathbf{R}),\sigma^2|\boldsymbol{m},\boldsymbol{\Sigma},a,b) &(4.60)\\ &= \mathcal{N}(\mathrm{vec}(\mathbf{R})|\boldsymbol{m},\sigma^2\boldsymbol{\Sigma})\mathcal{G}^{-1}(\sigma^2|a,b) &(4.61)\end{aligned}$$

By letting $\boldsymbol{m} = \text{vec}(\mathbf{M})$ and $\boldsymbol{\Sigma} = \mathbf{V} \otimes \mathbf{U}$, we may express the prior as:

$$\begin{aligned}
\Pr(\mathbf{R}, \sigma^2) &= \mathcal{NG}^{-1}(\text{vec}(\mathbf{R}), \sigma^2 | \text{vec}(\mathbf{M}), \mathbf{V} \otimes \mathbf{U}, a, b) && (4.62)\\
&= \mathcal{N}(\text{vec}(\mathbf{R}) | \text{vec}(\mathbf{M}), \sigma^2 \mathbf{V} \otimes \mathbf{U}) \mathcal{G}^{-1}(\sigma^2 | a, b) && (4.63)
\end{aligned}$$

We can see that the multivariate normal component of this prior has an identical form to that of Equation 4.57, and hence we are able to imbue $\mathbf{R}$ with the properties of a full matrix-variate normal distribution by choosing our hyperparameters a, b, $\mathbf{M}$, $\mathbf{U}$ and $\mathbf{V}$ appropriately such that they are informed by the structures of the visual and semantic spaces.

Often it is the case that there is no special prior knowledge of $\mathbf{M}$ governing the mean of the prior ratings and so $\mathbf{M} = \mathbf{0}$ (also simplifying some of the associated mathematics). However, it is often possible to craft a $N \times N$ matrix of item covariances, $\mathbf{S}$, or a $P \times P$ matrix of attribute covariances, $\mathbf{T}$, based on knowledge of the problem structure, and so useful to set $\mathbf{U} = \mathbf{S}$ and $\mathbf{V} = \mathbf{T}$. For the aforementioned Netflix example, we may assume films from similar genres receive similar ratings, or that viewers from similar demographics issue similar ratings. Likewise, we can expect texture attributes from the same cluster (see Section 3.2.2) to be reasonably similar in sentiment, and so for attributes i and j from the same cluster we could set $\mathbf{T}_{ij}$ to some positive value indicating that fact. In this way we are able to model more completely the underlying structure of the attribute strength matrix and to make better estimates of its values when provided with only modest numbers of pairwise comparisons.

The derivation of the posterior distribution follows in an identical manner as for the single attribute case (Equation 4.40):

$$\Pr(\mathbf{R}, \sigma^2 | \boldsymbol{y}) = \mathcal{NG}^{-1}(\text{vec}(\mathbf{R}), \sigma^2 | \text{vec}(\mathbf{M}_*), \boldsymbol{\Sigma}_*, a_*, b_*) \tag{4.64}$$

with posterior hyperparameters obtained analogously to Equation 4.42. Likewise, the marginal posterior distribution over the item ratings is derived as for the single attribute case (Equation 4.46):

$$\Pr(\mathbf{R} | \boldsymbol{y}) = \mathcal{T}(\text{vec}(\mathbf{R}) | \text{vec}(\mathbf{M}_*), \frac{b_*}{a_*} \boldsymbol{\Sigma}_*, 2a_*) \tag{4.65}$$

The mode of the posterior distribution over $\mathbf{R}$ thus follows from Equation 4.51, and by making use of the identity in Equation 4.3 can be expressed as:

$$\text{vec}(\hat{\mathbf{R}}^{\text{MAP}}) = (\mathbf{V}^{-1} \otimes \mathbf{U}^{-1} + \mathbf{C}^\top \mathbf{C})^{-1} (\text{vec}(\mathbf{U}^{-1} \mathbf{M} \mathbf{V}^{-1}) + \mathbf{C}^\top \boldsymbol{y}) \tag{4.66}$$

We note that this estimate requires the inversion of a matrix whose dimensions are quadratic in both N and P. For large N and P we are required to seek ways of attaining

$\hat{\mathbf{R}}^{\mathrm{MAP}}$ without explicitly calculating the Kronecker product between $\mathbf{U}$ and $\mathbf{V}$ or the block-diagonal matrix C. A discussion of such techniques is given in Appendix B.

4.3.3 Generalisation

Now that a probabilistic model has been defined, in this section we briefly discuss means by which the outputs of the model can be used to generalise to new items and attributes.

As mentioned previously, the most natural means of doing this within the probabilistic formulation is by updating $\mathbf{S}$ (for items) or $\mathbf{T}$ (for textures) to reflect the similarity of the new items or attributes to those that have already been observed in the pairwise comparison data, and to then re-evaluate the posterior distribution with these updated priors. Although this allows generalisation across both textures and attributes, it is computationally expensive to calculate the posterior, prohibitively so when large numbers of unseen items must be rated.

Instead, we may desire our model to output a *ranking function* as part of the learning phase, which can then be used independently without requiring the training data again. For the (linear) Rank SVM trained on comparison data for attribute p, this output takes the form of a vector $\boldsymbol{w}_p$ which can be applied to a feature space representation of the items to obtain rating estimates:

$$\hat{\boldsymbol{r}}_p = \mathbf{X}\boldsymbol{w}_p \tag{4.67}$$

In contrast, for the probabilistic formulation in this chapter $\hat{\boldsymbol{r}}_p$ is the primary output of the model. In this context, solving the system of linear equations above to find the unknown value $\boldsymbol{w}_p$ leads to:

$$\boldsymbol{w}_p = \mathbf{X}^{\dagger}\hat{\boldsymbol{r}}_p \tag{4.68}$$

This makes use of the Moore-Penrose pseudoinverse and so $\boldsymbol{w}_p$ represents a least squares solution to the linear system (Penrose, 1956). It applies equally to a matrix ranking function over multiple attributes:

$$\mathbf{W} = \mathbf{X}^{\dagger}\hat{\mathbf{R}} \tag{4.69}$$

However, these least squares solutions neglect to make use of the full probabilistic distribution over $\hat{\mathbf{R}}$, only working on a single estimate.

Alternatively, we may seek to obtain a distribution over a particular ranking function $\Pr(\mathbf{w}_p|\boldsymbol{y}_p)$ as the primary learned output of the model, such that we can directly quantify the likelihood of a particular ranking function, or marginalise over the space of ranking functions when evaluating ratings. Such an approach could proceed by noting again Equation 4.67, and so can perform this substitution in most of the distributions and estimates outlined in this chapter. In this way, for instance, we can acquire a posterior distribution $\Pr(\mathbf{w}_p|\boldsymbol{y}_p)$ and an associated MAP estimate:

$$\hat{\boldsymbol{w}}_p^{\mathrm{MAP}} = (\boldsymbol{\Sigma}^{-1} + \mathbf{X}^\top \mathbf{C}^{(p)\top} \mathbf{C}^{(p)} \mathbf{X})^{-1} (\boldsymbol{\Sigma}^{-1} \boldsymbol{m} + \mathbf{X}^\top \mathbf{C}^{(p)\top} \boldsymbol{y}_p) \tag{4.70}$$

However, when extending this to the matrix normal setting the support is no longer over a $N \times P$ matrix of ratings, but over a $N \times D$ matrix of ranking function coefficients, and so the between-column prior covariance matrix represents associations between ranking function coefficients rather than attributes. It is therefore not as simple to incorporate $\mathbf{T}$, our knowledge of the attribute space structure, and so we lose the benefit of jointly learning multiple attributes.

4.3.4 Summary

In this section we derived a probabilistic approach to inferring attribute strengths from pairwise texture comparison based on the multivariate Gaussian distribution. This approach was designed with the texture dataset described in the previous chapter in mind, which we know to be both sparse (only a small percentage of the total possible labels were obtained) and likely featuring a non-negligible proportion of mislabellings due to it being crowdsourced from remote workers with a financial incentive to work quickly.

To combat the sparsity we employed a Bayesian methodology, allowing us to define prior distributions over $\mathbf{R}$ in order to influence the form of the posterior distribution $\Pr(\mathbf{R}|\boldsymbol{y})$ based on knowledge of the attribute and texture spaces. The fewer the number of comparisons provided as evidence when calculating the posterior distribution, the greater the effect the prior distribution has in avoiding overfitting. Specifically, we describe how similarity matrices over the texture and attribute spaces may be constructed, how these can be used to inform the prior covariance, and how they can be used to allow generalisation over both attributes and items. We assess the resilience of the model to sparsity in Section 4.4.2, and its resilience to mislabelling in Section 4.4.3.

Next, to deal with erroneous labels from respondents the derivation explicitly models the error associated with labels to be zero-mean normally-distributed homoscedastic noise (Equation 4.23). We test how the model performs under varying levels of Gaussian noise in Section 4.4.4.

Finally but importantly, because the primary learned output of the model is a probability distribution, we are able to acquire a deeper understanding of the ratings and associated data than with a single estimate, including the ability to predict future comparisons (Equation 4.53), to estimate model error (Equation 4.52), and to quantify how certain a particular rating was.

	RankSVM	**MVN MLE**	**MVN Bayes**	**Matrix Bayes**
Section	Section 4.2.2	Section 4.3.1.1	Section 4.3.1.2	Section 4.3.2
Inputs	$\mathbf{X}$	—	$\mathbf{S}$ (and hyperparameters)	$\mathbf{S}, \mathbf{T}$ (and hyperparameters)
Ratings	Obtain $\boldsymbol{w}$ using Equation 4.20 $\boldsymbol{r} = \mathbf{X}\boldsymbol{w}$	Equation 4.33	Equation 4.51	Equation 4.66
Rating distribution	—	—	Equation 4.46	Equation 4.65
Rating new item	Obtain feature vector $\boldsymbol{x}$ $r = \boldsymbol{x}^\top \boldsymbol{w}$	—	Update $\mathbf{S}$ and re-calculate posterior (Section 4.3.3)	Update $\mathbf{S}$ and re-calculate posterior (Section 4.3.3)
Rating new attribute	—	—	—	Update $\mathbf{T}$ and re-calculate posterior (Section 4.3.3)

Table 4.1: Key methods and equations for (i) the Ranking SVM, (ii) the maximum likelihood estimate of the multivariate normal distribution, (iii) the Bayesian treatment of the multivariate normal distribution, and (iv) the Bayesian treatment of the matrix-variate normal distribution.

However, there are some drawbacks to the probabilistic methodology. Primarily, it is far more computationally expensive to compute the posterior distribution or its related estimates than it is to solve the optimisation problem inherent to the Rank SVM. Calculation of the full MAP estimate (Equation 4.66) for a large problem with dense $\mathbf{U}$ and $\mathbf{V}$ involves K iterations of the conjugate gradient method taking $\mathcal{O}(N^2P^2K)$ time, where K is the condition number of $\mathbf{V}^{-1} \otimes \mathbf{U}^{-1} + \mathbf{C}^\top\mathbf{C}$ (Saad, 2003). In comparison solving the primal form of the traditional Rank SVM for P attributes can be done in time $\mathcal{O}(PND + PN \log N)$ (Chapelle and Keerthi, 2010). This makes the probabilistic method intractable for large problems that the Rank SVM can handle easily.

For comparison, the key features of each approach is shown in Table 4.1. In the next section we test the assumptions of our model and assess the performance of these learning approaches across a variety of tests using synthetic data.

4.4 Evaluation

We generated a synthetic dataset to test the techniques described in the previous sections under controlled conditions. This allows us to easily control for varying numbers and qualities of pairwise comparison responses, and to test the impact of the dataset's between-item and between-attribute correlation structure. For each test, we compared the rating estimates given by the Ranking SVM (`Rank`, Equation 4.20), the multivariate normal maximum likelihood estimate (`MLE`, Equation 4.33), and the matrix-variate normal MAP estimate (`MAP`, Equation 4.66). Following is a description of how the synthetic dataset was generated and how results were collected.

4.4.1 Experimental setup

Results were averaged over 1000 iterations for each experiment: in all cases standard errors were sufficiently low for conclusions to be drawn unambiguously. The number of items, N, their feature vector dimensionality, D, the number of attributes, P, and the number of comparisons, M, were all variables defined for each experiment and constant between iterations.

Data generation

At each iteration the synthetic dataset was regenerated in the following manner:

- Between-item correlations $\mathbf{S}$ were randomly generated using the *vine* method described in Lewandowski et al. (2009) with $\eta = 1$,
- A $(P + D) \times (P + D)$ correlation matrix is generated, $\mathbf{A}$, again using the method above with $\eta = 1$. Between-attribute correlations $\mathbf{T}$ are taken as the top-left $P \times P$ sub-matrix, between-feature correlations $\mathbf{F}$ are taken as the bottom-right $D \times D$ sub-matrix. In this way unknown correlations are induced between attributes and features, as with real data.
- An $N \times (P + D)$ matrix was sampled from a zero-mean matrix-variate normal distribution with between-row and between-column covariance matrices set to $\mathbf{S}$ and $\mathbf{A}$ respectively. The first P columns of this matrix constitute $\mathbf{R}$, and the latter D columns constitute $\mathbf{X}$,
- The variance of the subject error was sampled from an inverse gamma distribution: $\sigma^2 \sim \mathcal{G}^{-1}(1, 1)$.

In this way the generated data provides $\mathbf{S}$, $\mathbf{T}$, $\mathbf{X}$ as input for the models, and $\mathbf{R}$, the unknown rating matrix, defined such that its items and attributes share an underlying covariance structure with $\mathbf{X}$.

Comparison generation

A comparison was then generated for each attribute in turn, with the two items chosen randomly each time, until all M comparisons had been done. The comparison outcome was sampled as if from the distribution in Equation 4.26, and then truncated to be either 1 or -1 so as to imitate genuine pairwise comparison data.

Model parameters

The `MAP` estimate is a function of $\mathbf{\Sigma}$ and $\mathbf{M}$, hyperparameters of the Bayesian prior over the ratings. We set $\mathbf{\Sigma} = \mathbf{T} \otimes \mathbf{S}$ and $\mathbf{M} = \mathbf{0}$, as set out in Section 4.3.2. For `Rank`, the trade-off between maximising the margin and minimising the misclassification error is set as $C = 1$.

Evaluation function

Each estimation technique was then trained on these pairwise comparison results using the relevant equation in Table 4.1, and evaluated by calculating the mean Spearman rank correlation between the estimated and actual item ratings across all attributes. The Spearman correlation is appropriate because it is only concerned with the monotonic relation between two variables and the attribute strength ratings are essentially scaleless. For estimated ratings $\hat{\mathbf{R}}$ and actual ratings $\mathbf{R}$ the mean Spearman correlation ρ is:

$$\rho = \frac{1}{P} \sum_{p=1}^{P} \left(1 - \frac{6 \sum_i^N (\hat{\mathbf{R}}_{ip}^{\text{RANK}} - \mathbf{R}_{ip}^{\text{RANK}})^2}{N^3 - N} \right)$$

where the superscript $^{\text{RANK}}$ denotes the *rank* of the specified rating within each attribute, rather than the raw value.

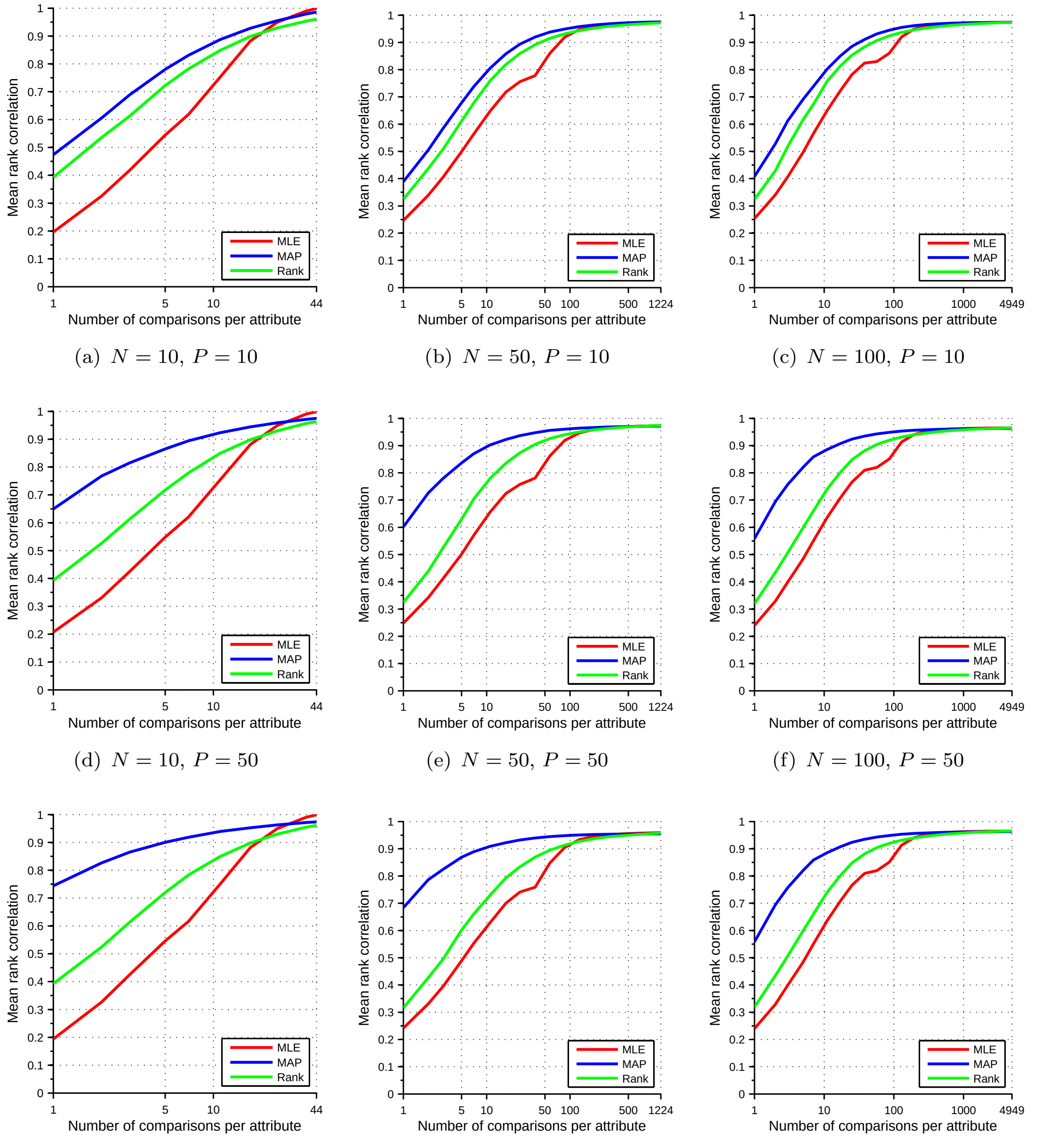
We begin in the section by investigating the effect of varying numbers of pairwise comparisons.

4.4.2 Effect of number of pairwise comparisons

We first investigate the effect of varying M for fixed values of N, P and D. Given values for N and P, the total number of unique pairwise comparisons that can be made is $P \times \binom{N}{2}$. In Figure 4.1 results are shown for combinations of $N = \{10, 50, 100\}$ and $P = \{10, 50, 100\}$, with $D = 10$ for all experiments. A logarithmic scale is used for the x-axis so that results are emphasised for smaller values of M — we expect this to be the prevailing situation when collecting pairwise comparison data.

In all cases the Bayesian probabilistic `MAP` estimate dominates over both the ordinary maximum likelihood estimate `MLE` as well as `Rank`. This advantage is more pronounced for increasing values of P due to the incorporation of the between-attribute similarities $\mathbf{T}$ into the normal-inverse-gamma hyperparameters. When the correlations held in $\mathbf{T}$ are suitably strong even very lower numbers of attributes per comparison can deliver good results, as they effectively contribute to the ratings of all sufficiently related attributes. This can be seen, for example, for $N = 50$, $P = 100$, where just ten comparisons per attribute are sufficient for `MAP` to achieve a 90% mean rank correlation while `Rank` is only able to attain 70%. However, ordinary maximum likelihood estimation only wins out over `Rank` for the largest values of M, indicating that if no particular prior knowledge is known for the distribution of **R** and a probabilistic interpretation is not required then the Ranking SVM is still preferred.

Finally, inspection of the graphs provides more evidence that only relatively low numbers of comparisons are needed to reach near-optimal performance for a given estimator, in the order of N comparisons per attribute. This corroborates findings reported in Parikh and Grauman (2011).



(a) $N = 10$, $P = 10$

(b) $N = 50$, $P = 10$

(c) $N = 100$, $P = 10$

(d) $N = 10$, $P = 50$

(e) $N = 50$, $P = 50$

(f) $N = 100$, $P = 50$

(g) $N = 10$, $P = 100$

(h) $N = 50$, $P = 100$

(i) $N = 100$, $P = 100$

FIGURE 4.1: Mean Spearman rank correlation coefficients across all attributes, calculated for varying M with P and N held fixed. x-axes are shown with a logarithmic scale to emphasise lower values of M.

4.4.3 Effect of similarity strength

Because the probabilistic methods we have derived operate on similarity matrices, we can expect their performance to deteriorate when correlations are weakened and *vice versa*. As observed in the previous experiment, `MAP` performs better than `Rank`, but to what extent does the correlation structure have to break down for it to approach the performance of `MLE`? For the synthetic dataset, random positive-semidefinite correlation matrices are generated using the *vine* method of Lewandowski et al. (2009). Here, partial correlations are generated between 0 and 1 for each pair of indices, which are then recursively formed into a full positive-semidefinite correlation matrix. We generate each partial correlation such that it is drawn from a Beta(η, η) scaled and translated to have support from -1 to 1, where η dictates the strength of the partial correlations and takes values of 2^k for $k = 0, \ldots, 7$. Probability density functions for selected values of η are shown in Figure 4.2, where it can be seen that progressively larger values of η bias the distribution towards small correlations. For non-synthetic datasets the equivalent η value can be estimated by fitting a Beta distribution to the non-diagonal elements of the correlation matrix.

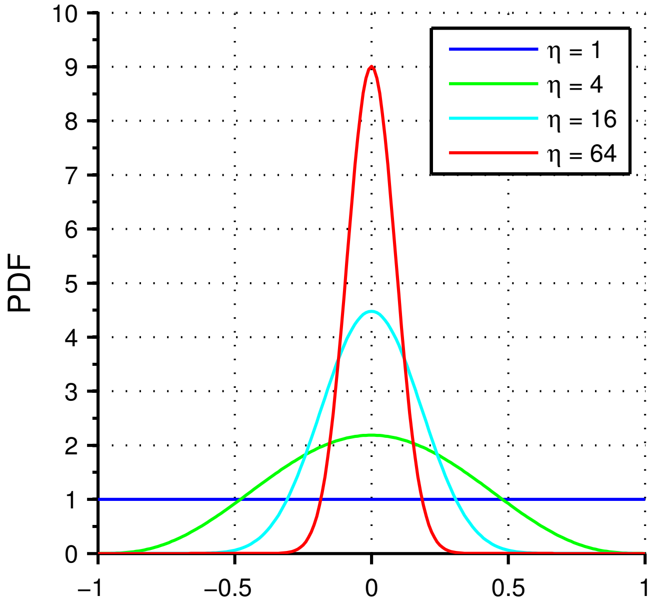
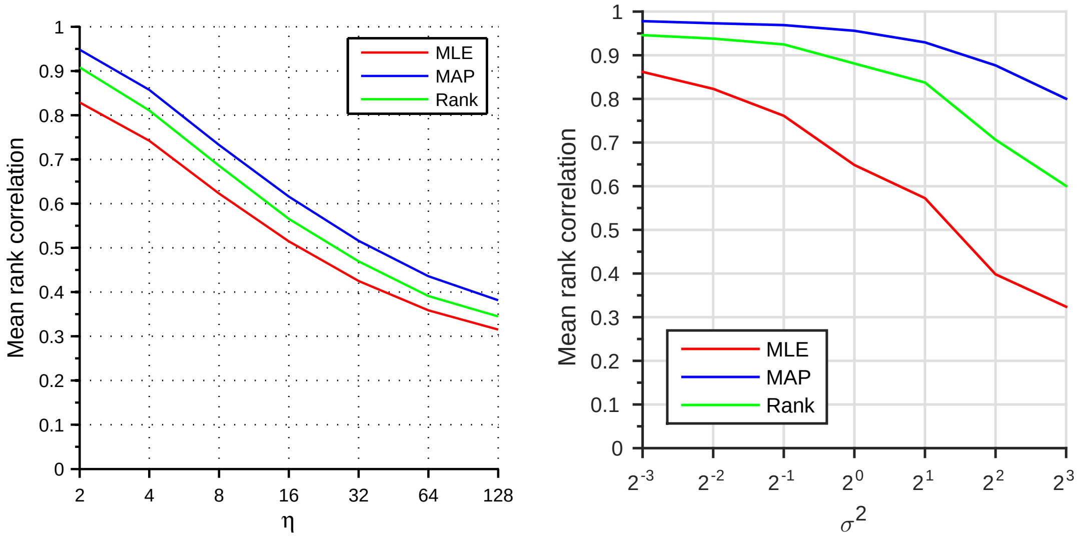


FIGURE 4.2: Probability density functions of Beta(η, η) scaled and translated to support -1 to 1 for selected values of η.

The test was conducted with $M = N = P = 50$. The results in Figure 4.3(a) show each technique suffers roughly equally when the level of correlation is reduced. Although `MAP` may have been expected to be particularly sensitive to the effects of dataset correlations due to it depending directly upon them, it can be seen that both `MLE` and `Rank` suffer proportionately due to the greater likelihood of learnt ranking matrices being unable to generalise to unseen data. For higher values of η this unseen data has a greater probability of being unrelated to previous observations, an effect which impacts all learning algorithms equally. Thus, the correlation structure of the dataset needn't influence which technique is chosen so long as the prior covariance over $\mathbf{R}$ properly reflects that structure.



(a) Varying η

(b) Varying σ^2

FIGURE 4.3: Mean Spearman rank correlation coefficients across all attributes, calculated for varying η (left) and σ^2 (right) with $M = N = P = 50$.

4.4.4 Effect of respondent reliability

In real-world situations the pairwise comparison data elicited from respondents will be prone to subjective error. This error may be due to the respondent's personal interpretation of the attribute meaning, cognitive bias associated with the pairwise methodology, or even simply boredom and lack of interest. It was one of our stated goals that the learning procedure should be robust to noise in $\boldsymbol{y}$, and here we test the degree to which each technique is resilient to erroneous responses.

In the synthetic dataset we can influence the degree of respondent error by changing the value of σ^2, the variance of the corruption to each pairwise comparison response. With higher levels of corruption respondents are more likely to deliver erroneous responses as their perception of each attribute strength has a greater probability of deviating considerably from the mean (the actual attribute strength). We tested for $\sigma^2 = 2^k$ for $k = -3, \ldots, 3$ and with $M = N = P = 50$; results are shown in Figure 4.3(b).

It can be seen that for very low values of σ^2 — right down to when $\sigma^2 = 0$ and there is no subjective error — `MAP` and `Rank` perform fairly similarly. However, the former's performance deteriorates more gradually, only decreasing 0.18 over the course of the values shown, compared to 0.35 for `Rank`, and 0.53 for `MLE`. This effect is due to the Bayesian prior over the attribute strength ratings softening the influence of data which contradicts these priors. Hence a probabilistic approach with well-informed priors is advantageous when it is suspected that respondents may give sub-optimal data, and furthermore provides a natural framework in which individual respondent reliability can be tested and possibly discarded. We describe an experiment to do this in Section 5.1.

4.5 Summary

In this chapter we derived a new Bayesian probabilistic framework for learning from pairwise comparison data. Being probabilistic, this framework has the natural advantage that it can provide a principled measure of uncertainty and — in a Bayesian setting — incorporate prior knowledge about the semantic and visual spaces into the learning procedure, allowing it to perform well with fewer numbers of comparisons and in the face of considerable error in the pairwise comparison labels. When evaluated on synthetic data we demonstrated that the Bayesian maximum *a posteriori* estimate attained better performance than the popular Ranking SVM, especially when applied to sparse or noisy data. However, the time complexity of computing the `MAP` estimate of the Bayesian posterior is quadratic in both N and P but linear for the Ranking SVM, which is a better choice for large problems or problems where there is no particular prior knowledge about the ratings structure.

The next chapter takes the learning procedures described above and uses them on the label dataset described in Chapter 3 to perform a range of experiments.

Chapter 5

Experiments

In this chapter we combine the labelling created in Chapter 3 with the learning procedure derived in the previous chapter to investigate the visual and semantic spaces of texture further:

- We use our Bayesian posterior distributions to evaluate the reliability of respondents to our labelling exercise.
- We appraise numerous visual texture descriptors in terms of how well they align with human perception of texture.
- We conduct a retrieval experiment to demonstrate how normal visual descriptors may be *semantically enriched* by combining them with our label dataset.

We begin with the evaluation of respondent reliability in the labelling task.

5.1 Respondent reliability

As described in Section 3.3 our pairwise comparison dataset was crowdsourced and unsupervised. Subjects received no assistance beyond help text accompanying each comparison and, furthermore, they possessed a monetary incentive to answer questions quickly. For this reason it is important to analyse the dataset and check for and discard any poor-quality responses.

Previous work has addressed this by designing redundancy into the elicitation procedure and then deferring to majority vote for each comparison point (Whitehill et al., 2009); using information theory (Simpson et al., 2013); or by minimising a global cost function measuring ranking inconsistencies (Fu et al., 2016). However, the probabilistic framework from the last chapter actually provides us with a natural setting in which to quantify the value of each subject's responses in a principled way due to its ability to measure the degree of respondent error present in the dataset. This can be done using the MAP estimate of the per-response error variance, $\hat{\sigma^2}^{\mathrm{MAP}}$, as in Equation 4.52. Here, $\hat{\sigma^2}^{\mathrm{MAP}}$ is an estimate of the variance of the response error under the assumption responses are corrupted by normally distributed noise.

Calculating across all comparisons in the dataset, the error variance is found to be $\hat{\sigma^2}^{\mathrm{MAP}} = 0.3065$. Next, for each subject in turn who submitted responses we again calculate the error variance across all comparisons *omitting* that subject. This gives an assessment of the level of error present in the dataset had that subject not been involved. By comparing this value to the value calculated across the whole dataset above we can calculate the impact each respondent has had on total response noise. To be precise, for an estimated variance of $\hat{\sigma^2}_i^{\mathrm{MAP}}$ obtained when omitting respondent i who made m_i comparisons, we can calculate the respondent's mean per-comparison impact on the total error, e_i, as:

$$e_i = \frac{\hat{\sigma^2}_i^{\mathrm{MAP}} - \hat{\sigma^2}^{\mathrm{MAP}}}{m_i} \tag{5.1}$$

Here, positive values indicate that there was greater total error variance *without* the respondent, and *vice versa* for negative values.

A bar graph showing the e values for all 568 respondents is shown in Figure 5.1. To aid in visualisation, subjects have been sorted in ascending order by their e scores — that is, subjects with scores above zero can be said to have had a positive impact on overall error variance, and *vice versa.*

We may look to use these values to cull unhelpful responses from the dataset, possibly because the respondent did not understand the attribute, or did not properly consider the comparison in order to progress through them quicker and earn more money. However, simply removing responses by all respondents with negative scores fails to distinguish between genuinely biased comparisons and those which by chance exhibit a natural variance due to the nature of the attributes or textures they feature.

To determine the optimal cut-off point at which to cull respondents we performed a four-fold cross-validation classification procedure calculated over the range of e values. Attribute strength ratings were calculated from the training set, and the misclassification rate was measured across all of the dominance relations in the holdout set. The method of David (1987) (see Section 4.2.1 for details) was used to calculate the attribute strength ratings, as it does not require any explicit feature space representation and operates directly on the pairwise comparison data. A graph plotting this data is shown in Figure 5.2.

From this graph it can be seen that for the very highest cut-off values the misclassification rate is high — barely better than random. This is because these cut-off values are overly

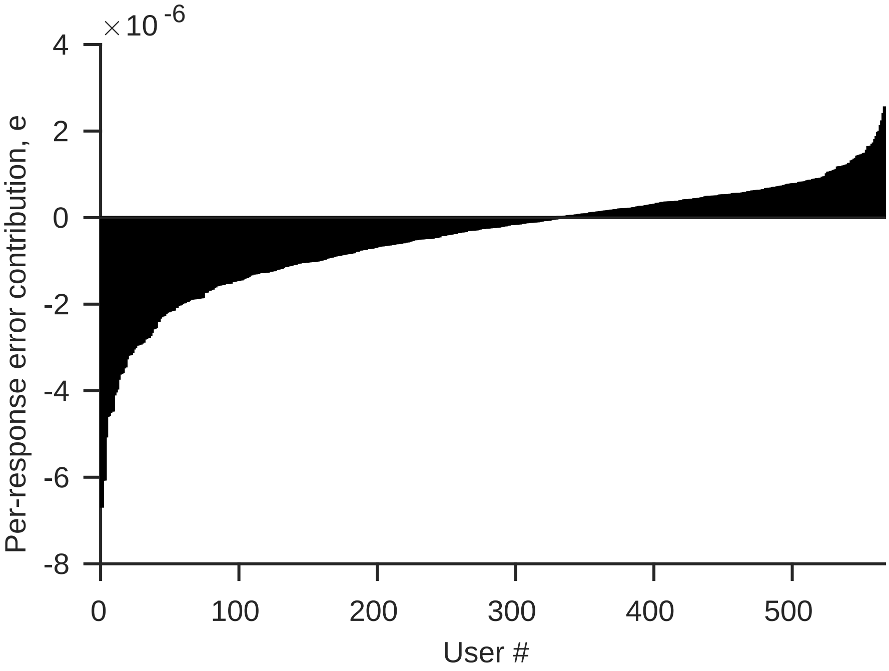


FIGURE 5.1: Each subject's mean contribution towards the error variance reduction. Subjects are sorted in ascending order of their contribution for ease of visualisation.

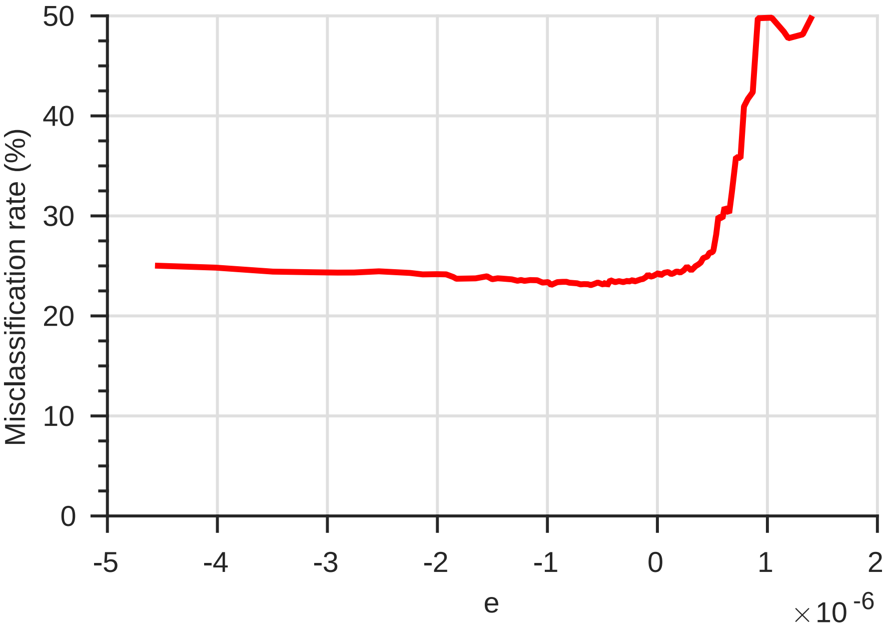


FIGURE 5.2: Misclassification rate calculated for different cut-off values of e when respondents with a value below the cut-off are omitted from the dataset.

restrictive, with only very few comparisons retained in the resulting datasets. As the cut-off is lowered and more comparisons are included the misclassification rate quickly drops, until it reaches a minimum at around -1×10^{-6}. Beyond this point the misclassification rate gently rises again, as lower quality comparisons are included by respondents with the most negative effect on the total error variance. With a cut-off of -1×10^{-6} there are 152 respondents and just under 25,000 pairwise comparisons excluded from the resultant dataset, leaving a total of 115,204 comparisons by 416 respondents – about 83% of the size of the original dataset.

In this section we have demonstrated a way of pre-processing pairwise comparison data in order to identify and remove noisy responses probabilistically using the Bayesian MAP estimates derived in the previous chapter. The technique is independent of any feature space representation, operating directly on the pairwise comparison graph, and can be naturally extended to operate *online* as results arrive, so that unhelpful respondents can be identified quickly and discarded, improving time and cost effectiveness.

5.2 Semantic appraisal of visual descriptors

In this section we appraise each of the texture descriptors introduced in Section 2.3 in terms of how well they reflect the structure of the semantic comparison graph for each of the ninety eight attributes and, therefore, how well they capture human perception of these attributes. These results allow us to identify regions within the semantic space of texture which are poorly modelled by current techniques, as well as the visual features which correspond best with human perception and which will provide the basis for our semantically-enriched descriptors.

5.2.1 Visual features to be assessed

Five different texture descriptors were assessed. For more detail on each descriptor refer to Section 2.3:

- Co-occurrence matrices (Haralick, 1979) are calculated for points situated along the perimeters of circles of radii 1, 2, 4, 8, and 16 pixels. Each of these five matrices are summarised in terms of their contrast, homogeneity, uniformity, entropy, variance, and correlation, resulting in a 30-element feature vector. (`CoM`)

- The mean and standard deviation of the Gabor wavelet responses of 24 orientation and scale combinations given in Manjunath and Ma (1996), yielding a feature vector of 48 elements. (`Gab`)

- The 8 optimal Liu noise-resistant features of the Fourier transform (Liu and Jernigan, 1990). (`Liu`)

- 16-dimensional feature vector derived from the Statistical Geometrical Features procedure (Chen et al., 1995) being performed at 31 regularly-spaced threshold levels. (`SGF`)

- Uniform (local) binary patterns (Ojala et al., 2002b) calculated for eight points around circles of three different radii: 2, 4, and 8. This gives a total concatenated feature vector of dimension 30. (`UBP`)

In the next section we describe the methodology used to assess these five descriptors.

5.2.2 Methodology

The semantic correspondence of each of the visual descriptors described above is evaluated using a 4-fold cross-validation procedure. For this experiment we are assessing each visual descriptor in terms of how well it is able to detect each individual attribute. For this reason the attribute strength ratings $\hat{\mathbf{R}}$ are learned from the training images using the Ranking SVM estimate shown in Equation 4.66, as we do not require the probabilistic features of the multivariate normal estimators.

The free parameter in the Ranking SVM equation, C, is allowed to vary between 21 logarithmically spaced values $(4^{-10}, 4^{-9}, \ldots, 4^{9}, 4^{10})$, the optimal value of which is selected through the cross-validation procedure.

A per-attribute ranking of all 319 textures is then derived from $\hat{\mathbf{R}}$. The misclassification rate is calculated over all dominance comparisons involving at least one of the textures in the hold-out set by simply comparing the rankings of the respective textures.

5.2.3 Results and analysis

Misclassification rates for each combination of descriptor and attribute are shown in Table 5.1, while average rates for each combination of descriptor and attribute cluster are shown in Table 5.2.

One of the descriptors best able to capture the structure of the semantic comparison graph was the Liu descriptor, comprising frequency measures based on the Fourier transform. It performed well for the two clusters involving regular placement of linear texture primitives: cluster I (`linear orientation`) and cluster II (`weave-like structure`). Inspection of the learned ranking function reveals that a high moment of inertia and low proportion of energy for the first quadrant of the normalised Fourier transform are the pertinent features for these two attributes. The Liu descriptor is also amongst the best performers for the polar notions of order (cluster IV) and disorder (cluster V): here, the inertia and energy of the first quadrant is again decisive. Low inertia and high energy indicates random texture while the opposite aligns more closely with repetitive texture.

The Gabor descriptor — like the Liu descriptor, based on a Fourier transform — also performed well, but appeared to be more resilient to disorder than the Liu descriptor.

Cluster I — Linear orientation

Attribute	CoM	Gab	Liu	SGF	UBP
Banded	0.29	0.23	**0.18**	0.36	0.23
Corrugated	0.36	**0.33**	**0.33**	0.34	0.36
Furrowed	0.34	**0.30**	0.35	0.34	0.31
Grooved	0.39	0.33	**0.28**	0.37	0.29
Lined	0.29	0.18	**0.15**	0.32	**0.15**
Pleated	0.31	0.25	**0.23**	0.34	0.26
Ribbed	0.34	0.26	0.23	0.32	**0.22**
Ridged	0.34	**0.29**	0.33	0.35	0.31
Stratified	0.37	0.36	**0.29**	0.39	0.34
Striated	0.35	0.28	**0.27**	0.38	**0.27**
Zigzagged	0.33	**0.29**	0.34	0.32	0.30

Cluster II — Weave-like structure

Attribute	CoM	Gab	Liu	SGF	UBP
Chequered	0.26	0.26	**0.25**	0.29	0.33
Cross-hatched	0.32	0.26	**0.23**	0.32	0.26
Fibrous	0.36	**0.32**	0.35	0.33	**0.32**
Gridlike	0.30	0.26	**0.20**	0.33	0.27
Honeycombed	**0.27**	0.28	0.32	0.28	0.31
Knitted	0.23	0.27	**0.21**	0.28	0.22
Matted	**0.39**	0.42	0.41	0.41	0.41
Meshed	**0.29**	0.31	0.32	**0.29**	0.34
Netlike	0.32	0.29	**0.27**	0.32	0.31
Waffled	0.31	0.31	**0.25**	0.33	0.30
Woven	0.28	0.24	**0.23**	0.29	0.24

Cluster III — Circular orientation

Attribute	CoM	Gab	Liu	SGF	UBP
Coiled	0.34	0.36	0.35	0.40	**0.32**
Corkscrewed	0.31	0.34	0.37	0.37	**0.28**
Flowing	0.30	**0.27**	0.31	0.31	0.29
Spiralled	**0.34**	**0.34**	0.36	0.38	0.37
Swirly	0.32	0.32	0.33	0.36	**0.30**
Twisted	**0.33**	**0.33**	0.40	0.38	0.39
Whirly	0.35	0.34	0.34	0.35	**0.33**
Winding	0.34	**0.33**	0.39	0.35	0.34

Cluster IV — Well-ordered

Attribute	CoM	Gab	Liu	SGF	UBP
Crystalline	**0.40**	**0.40**	0.41	0.42	**0.40**
Cyclical	0.32	0.34	**0.30**	0.34	0.32
Facetted	0.31	**0.30**	0.35	0.31	0.33
Fine	**0.23**	0.24	0.26	0.26	0.26
Harmonious	0.35	**0.30**	0.33	0.33	0.35
Lattice	0.32	0.28	**0.23**	0.35	0.30
Periodic	0.34	**0.31**	**0.31**	0.35	0.34
Regular	0.27	0.27	**0.25**	0.32	0.29
Repetitive	0.34	**0.30**	**0.30**	0.35	**0.30**
Rhythmic	0.34	0.29	**0.27**	0.37	0.31
Simple	**0.21**	0.23	0.23	0.24	0.24
Smooth	0.17	**0.16**	0.27	**0.16**	0.21
Uniform	0.26	**0.25**	0.26	0.29	0.29
Well-ordered	0.28	0.23	**0.19**	0.27	0.23

Cluster V — Disordered

Attribute	CoM	Gab	Liu	SGF	UBP
Asymmetrical	0.28	**0.22**	0.25	0.32	0.27
Complex	0.24	**0.23**	0.28	0.24	0.25
Discontinuous	0.24	0.25	**0.23**	0.28	0.26
Disordered	0.24	0.23	**0.22**	0.27	0.24
Indefinite	0.46	0.43	**0.36**	0.50	0.38
Irregular	0.24	**0.23**	**0.23**	0.29	0.25
Jumbled	0.21	0.20	**0.19**	0.24	0.22
Messy	0.21	**0.19**	0.20	0.25	0.23
Non-uniform	0.26	0.25	**0.24**	0.30	0.26
Random	0.27	0.23	**0.19**	0.29	0.23
Scattered	0.22	**0.18**	0.21	0.26	0.23
Scrambled	0.25	0.21	**0.20**	0.27	0.22

Cluster VI
Disordered circular primitives

Attribute	CoM	Gab	Liu	SGF	UBP
Dotted	0.31	**0.25**	0.35	0.27	0.27
Flecked	0.30	**0.25**	0.30	0.30	0.26
Freckled	0.30	0.26	0.29	0.28	**0.24**
Polka-dotted	0.32	**0.28**	0.37	0.31	0.30
Spattered	0.33	0.30	0.28	0.33	**0.27**
Speckled	0.30	0.27	0.27	0.29	**0.24**
Spotted	0.35	**0.31**	0.34	0.32	**0.31**
Sprinkled	0.26	0.23	0.23	0.28	**0.22**

Cluster VII
Disordered weave-like structure

Attribute	CoM	Gab	Liu	SGF	UBP
Braided	0.33	0.31	**0.25**	0.32	**0.25**
Cobwebbed	0.41	0.38	0.39	0.39	**0.37**
Entwined	0.33	**0.29**	0.31	0.36	0.31
Frilly	0.36	0.36	0.36	**0.35**	0.37
Gauzy	0.36	**0.30**	0.32	0.34	0.34
Interlaced	0.36	0.28	**0.26**	0.34	0.28
Intertwined	0.31	**0.28**	0.31	0.33	0.32
Lacelike	0.36	**0.30**	**0.30**	0.37	0.34
Webbed	0.34	0.28	**0.27**	0.35	0.35

Cluster VIII
Disordered indistinct circular primitives

Attribute	CoM	Gab	Liu	SGF	UBP
Blemished	0.35	0.38	**0.33**	0.35	**0.33**
Blotchy	0.32	0.32	0.31	0.38	**0.26**
Mottled	0.37	0.35	0.35	0.40	**0.30**
Smeared	0.30	0.32	0.34	0.30	**0.25**
Smudged	0.32	0.30	0.38	0.32	**0.28**
Stained	0.38	**0.31**	0.37	0.38	0.34

Cluster IX
Disordered linear primitives

Attribute	CoM	Gab	Liu	SGF	UBP
Cracked	0.30	**0.28**	0.30	0.29	0.32
Crinkled	**0.34**	0.35	0.36	0.35	0.39
Crow-footed	0.35	**0.32**	0.37	0.34	0.33
Fractured	**0.26**	0.27	0.27	0.30	**0.26**
Rumpled	0.37	0.35	**0.34**	0.35	0.38
Wizened	0.43	0.43	**0.38**	0.39	0.46
Wrinkled	0.37	**0.35**	0.36	0.40	0.38

Cluster X
Disordered indistinct linear primitives

Attribute	CoM	Gab	Liu	SGF	UBP
Marbled	0.31	0.28	0.36	0.32	**0.21**
Scaly	0.33	**0.30**	0.36	0.34	0.31
Veined	0.31	**0.27**	0.29	0.32	0.29

Cluster XI
Disordered circular 3D primitives

Attribute	CoM	Gab	Liu	SGF	UBP
Bubbly	0.28	0.25	0.28	0.28	**0.22**
Bumpy	0.23	0.22	0.31	**0.21**	0.25
Gouged	0.28	0.28	0.31	**0.27**	0.36
Holey	**0.30**	**0.30**	0.37	0.33	**0.30**
Perforated	**0.29**	**0.29**	0.36	0.30	0.34
Pitted	0.27	**0.26**	0.32	0.27	0.29
Porous	0.28	0.29	0.38	**0.27**	0.32
Potholed	0.27	**0.25**	0.28	0.29	0.29
Studded	0.28	**0.27**	0.34	**0.27**	0.28

Table 5.1: Misclassification rates for each combination of descriptor and attribute. Boldface denotes the strongest scoring descriptor for an attribute.

#	Cluster	CoM	Gab	Liu	SGF	UBP
I	Linear orientation	0.34	0.28	**0.27**	0.35	0.28
II	Weave-like structure	0.31	0.30	**0.28**	0.32	0.30
III	Circular orientation	0.33	**0.32**	0.35	0.35	0.33
IV	Well-ordered	0.29	**0.27**	0.28	0.31	0.29
V	Disordered	0.26	0.24	**0.23**	0.29	0.25
VI	Disordered circular primitives	0.31	0.27	0.30	0.30	**0.26**
VII	Disordered weave-like structure	0.35	**0.30**	0.31	0.35	0.32
VIII	Disordered indistinct circular primitives	0.34	0.33	0.34	0.36	**0.30**
IX	Disordered linear primitives	0.34	**0.33**	0.34	0.35	0.36
X	Disordered indistinct linear primitives	0.32	0.29	0.33	0.33	**0.27**
XI	Disordered circular 3D primitives	0.27	**0.26**	0.33	0.27	0.29

Table 5.2: Mean misclassification rates for each combination of descriptor and attribute cluster across all attributes in that cluster. Boldface denotes the strongest scoring descriptor for an attribute.

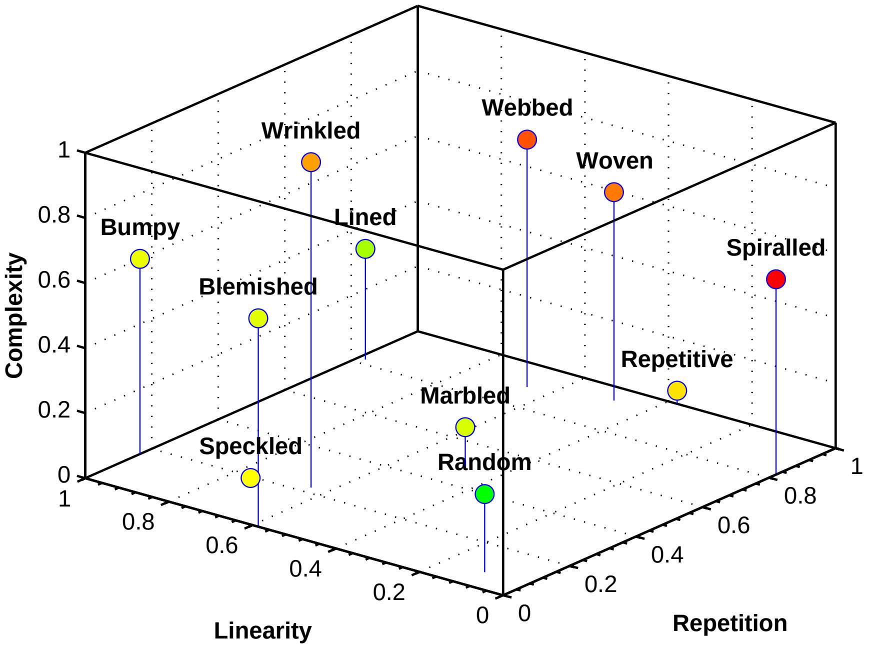


Figure 5.3: Sample attributes from each cluster plotted within the three-dimensional semantic space of Bhushan *et al*, coloured according to their scaled misclassification rate showing a deficiency in capturing attributes evoking complexity and disorder.

Indeed, it had among the lowest misclassification rates for clusters V through to XI, all associated with aspects of randomness. It has been speculated by Arcizet et al. (2008) that the primitive structures in human cognition of texture operate somewhat akin to Gabor filter, and our finding that a method based on them performs well on a large-scale dataset of human labels corroborates this.

The uniform binary patterns descriptor performs well for those attributes relating to disordered placement of small-scale primitives – clusters VI, VIII and X – as well as for another attribute associated with disorder, `marbled`. The uniform binary pattern descriptor is calculated as a histogram of local intensity patterns and so it is unsurprising that it performs relatively well for spatially-localised primitives such as speckles and bumps. By its nature the histogram makes no regard for placement rules making it better suited to capturing aspects of disorder. However, it is not amenable to deeper understanding as the local intensity patterns it detects have no immediately intuitive definition.

The two remaining descriptors, the co-occurrence matrix and statistical geometrical features, achieved good correspondence for certain attributes, but were invariably eclipsed by one of the other three descriptors.

Overall, the results indicate that there is considerable opportunity for improvement in the identification of visual features corresponding closely to human perception, especially for attributes exhibiting aspects of complexity or disorder such as in clusters VII to IX. Furthermore, textures involving strong elements of circular orientation were poorly assessed compared to those with linear orientation. These deficiencies are hardly unexpected, and tally with our knowledge of the workings of visual texture descriptors, but it is notable too that even correspondence with strongly regular attributes such as `chequered` and `repetitive` is only average.

Even despite the lack of correspondence between these human and machine interpretations of texture, semantic data may still be used to improve performance in tasks involving texture analysis. In the next chapter we demonstrate that semantic texture description results in considerable performance gains over a purely visual approach.

5.3 Content-based image retrieval

In this section we demonstrate the practical benefit of semantic data in a retrieval experiment.

5.3.1 Methodology

Each of the 3828 samples in the dataset is used in turn as a query texture against the remaining 3827 textures in the target set, of which only 11 are relevant to each query (each texture class has 12 samples due to variation in rotation and illumination). All textures in the target set are then sorted by the Euclidean distance of their descriptors from the query texture's descriptor yielding a ranking $\boldsymbol{r}$ where $r_i = 1$ if the member of the target set at rank i is relevant to the query, and 0 otherwise. This is done for all five descriptors introduced in Section 5.2.1.

For each descriptor a corresponding rating matrix was calculated using the MAP estimate given in Equation 4.66. That is, ratings for all 98 attributes are jointly learned from all comparisons involving textures in the test set. We then generalise to the unseen query texture by calculating a ranking function using the least-squares solution detailed in Section 4.3.3.

This ranking function was then applied to the texture features to create a new 98-dimensional *semantic descriptor* for each texture sample. As such, for each of the five visual descriptors we also have a corresponding set of semantic feature vectors.

Here, we set $\mathbf{M} = \mathbf{0}$. The 319 texture classes that the 3828 textures belong to are unknown in the setting of the retrieval experiment, and as such we assume no prior knowledge of the item covariance structure and set $\mathbf{U} = \mathbf{I}_N$. However, we impose a mild correlation structure on the attribute space by setting $\mathbf{V}_{ij} = 0.5$ for all attributes i and j which belong to the same cluster of texture words (see Table 5.1).

Lastly, we created a concatenated descriptor from all five visual descriptors which in turn allows another semantic descriptor to be learned from the most discriminative features across all descriptors. Again, the distances between the target set samples and the query sample are calculated for these concatenated and semantic descriptors, and a ranking derived.

From the relevance indicators of the n closest textures $(r_1, \ldots, r_n)$ for each query image we are able to calculate *precision* and *recall* measures, where precision is the proportion of the retrieved samples that are relevant, and recall is the proportion of the relevant samples that are retrieved:

$$\text{precision}(n) = \frac{\sum_{i=1}^{n} r_i}{n} \tag{5.2}$$

$$\text{recall}(n) = \frac{\sum_{i=1}^{n} r_i}{11} \tag{5.3}$$

Precision and recall are then calculated as n is allowed to vary from 1 to 3,827. We also calculate two summary measures of the ranked data:

- Mean average precision (MAP). The average precision (AP) for a given query is the average of the precision at each rank at which a relevant item is located:

$$\text{AP} = \frac{\sum_{i=1}^{3,827} r_i \, \text{precision}(i)}{11} \tag{5.4}$$

  This quantity is in turn averaged over all 3,828 queries.

- Equal error rate (EER). denoting the error rate at the point where the true positive rate (the recall) equals the false positive rate. It is calculated as the point on an ROC curve which intersects the diagonal connecting 100% on the x and y axes.

In the next section we discuss the results obtained when performing this methodology.

	MAP		EER	
Descriptor	**Visual**	**Semantic**	**Visual**	**Semantic**
CoM	19.4%	**42.0%**	18.2%	**8.8%**
Gab	21.4%	**35.8%**	20.3%	**11.8%**
Liu	18.9%	**23.6%**	22.7%	**12.8%**
SGF	**27.7%**	25.3%	**14.0%**	15.6%
UBP	**76.9%**	58.0%	5.6%	**5.4%**
Concatenated	43.6%	**64.4%**	9.7%	**3.0%**

Table 5.3: Mean average precision (MAP) and equal error rates (EER) for each descriptor across all 3,828 texture queries. Boldface denotes the highest scorer of each visual and semantic descriptor pair.

5.3.2 Results and analysis

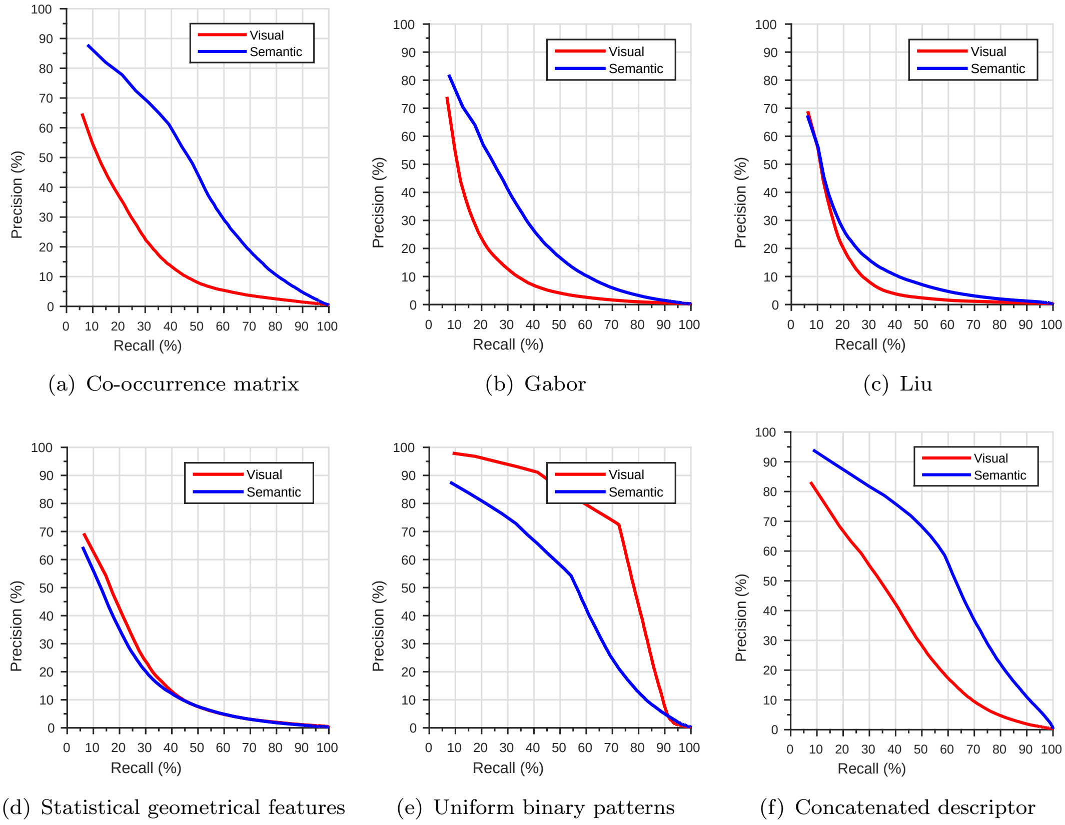
Precision-recall curves for both the visual and semantic version of each descriptor are shown in Figure 5.4. MAP and EER scores are shown in Table 5.3.

For four of the six curves it is immediately evident that the semantic descriptor gives higher retrieval performance than for the corresponding low-level visual descriptor, and very similar for one other, (SGF). The exception is uniform binary patterns, which was trained to perform well using the Outex dataset.

The benefit of the semantic descriptors is especially pronounced for higher rates of recall, where they often retrieve relevant textures with a considerably higher rate of precision. This improved precision at higher recall values could be interpreted as being indicative of greater robustness in the semantic descriptor: whereas the visual descriptors appear to struggle to recall all variations of rotation and illumination for a given query, the semantic descriptor is imbued with the invariant qualities that come from learning from the semantic comparison graph, and so generally is able to better recall variations of the same texture. This initial impression from inspecting the curves is reinforced upon viewing the summary values in Table 5.3: the semantic descriptor achieves higher EER scores in all cases but one. However, although the semantic form of the concatenated descriptor is the best overall descriptor in terms of EER, the visual form of the UBP descriptor is the best in terms of MAP.

It is worth noting that results could be further improved using a specialist retrieval metric based on probability, rather than Euclidean distance, by interpreting the semantic features not as point estimates, but as random matrices drawn from the multivariate t-distribution Equation 4.46. However, we have restricted our attention to Euclidean distance to enable a more like-for-like comparison.

Next, we summarise the experiments performed in this chapter.



(a) Co-occurrence matrix (b) Gabor (c) Liu

(d) Statistical geometrical features (e) Uniform binary patterns (f) Concatenated descriptor

Figure 5.4: Average precision-recall curves for each descriptor across all 3,828 texture queries. For each query there are 11 relevant and 3,816 irrelevant samples.

5.4 Summary

In this chapter we ran three experiments. The first used our probabilistic framework to prune unhelpful comparisons in a principled way, by assessing the subject's average contribution to total error variance. Next we demonstrated how visual descriptors could be assessed in terms of how well they represent the semantic space of texture, identifying which techniques work well and finding numerous regions of the semantic space which are captured poorly. Lastly, we ran a retrieval experiment to investigate the benefit of *semantically enriched* texture descriptors.

We conclude this thesis in the next chapter.

Chapter 6

Conclusions

In this thesis we presented the first approach to semantic texture characterisation and analysis, demonstrating that explicit semantic modelling provides considerable benefit when describing textures. Core to our approach was a ninety-eight word semantic space for texture, identified by Bhushan et al. (1997) as capturing much of the variability in human texture expression, and acting as low-level semantic attributes particularly well-suited to symbol grounding.

A psychologically-appealing pairwise comparison methodology was employed to elicit attribute strength ratings for 319 classes of visual texture, forming a large crowdsourced dataset of almost 140,000 pairwise comparison labels (Matthews, 2014) — to our knowledge, the largest of its kind. This methodology removes many of the cognitive biases associated with conventional data elicitation techniques, and the resultant dataset enables new semantically focused performance metrics to be used when assessing texture descriptors.

Pairwise comparison data presents a special challenge for inference, which must proceed using indirect evidence of the texture strengths in the form of comparisons, rather than through direct observations. Typical approaches to doing this offer no way of incorporating prior knowledge of the attribute strength structure, cannot measure uncertainty in the data, and must infer strengths for each attribute individually. To combat this we derived a new Bayesian probabilistic approach to learning from pairwise comparison data based on the matrix-variate normal distribution, allowing the specification of prior knowledge through Bayesian hyperparameters, a precise probabilistic quantification of uncertainty, and joint learning across *all* attributes. We demonstrated its effectiveness over a maximum likelihood approach and over the Ranking SVM, and showed how it could be used to measure the worth of the comparisons performed by each respondent to our crowdsourcing task, and to prune those that were not beneficial.

We then performed a semantic appraisal of existing visual texture descriptors, evaluating in a principled way how well they were able to model the ninety eight attributes we had selected. Numerous deficiencies were identified which saw the visual descriptors fail to capture the variability of the semantic space as held in the comparison responses. It is important that these deficiencies are addressed so as to properly bridge the semantic gap for texture and to pave the way for closer correspondence to human perception and expectations in user-centred visual applications.

Even despite this we demonstrated that an explicit semantic modelling step provides numerous benefits when describing textures. As well as allowing for more user-friendly interaction due to the bridging of the semantic gap, we demonstrated an improvement in retrieval rate for all but one of the visual descriptors tested. Furthermore, the use of attributes introduces a natural efficiency and robustness in the design of feature vectors, owing to the evolution of human language and the invariant qualities of human visual perception.

We hope in future to further improve our retrieval results and, consequently, to continue to close the semantic gap for texture. There are many avenues of exploration available beyond this initial demonstration of semantic modelling, some of which we detail in the next section.

6.1 Future work

It would be beneficial to repeat the lexicon creation procedure of Bhushan et al. (1997) with a probabilistic methodology. In this way, attributes would not occupy precise points within the semantic space of texture but probabilistic regions dictated by the consistency of subject responses. This data would allow a deeper analysis to be performed when subsequently learning from pairwise comparisons, and would better inform our between-attribute correlation structure held in $\mathbf{T}$. This also makes intuitive sense: it seems likely that a word like *bumpy* does not have the same unwavering definition from subject-to-subject, but varies according to individual perception and understanding.

Our probabilistic framework is also a natural setting in which to employ an *active learning* methodology, whereby data elicitation is dynamic and adaptive, with each pair of textures displayed to subjects according to some utility measure based on all the pairwise comparisons that went before it. For instance, it is of greater use to learn about the relationship between two textures for which the attribute strengths are quite uncertain, than it is to learn about the relationship between two very different textures for which we are already very certain of the ordering. Furthermore, respondents prone to submitting noisy data can be identified and excluded early in the elicitation process, rather than during pre-processing as we demonstrated in Section 5.1. These approaches allow data to be crowdsourced in a more efficient manner, achieving a greater coverage of the space of pairwise comparisons for the same cost, and improving the quality of subsequent inferences.

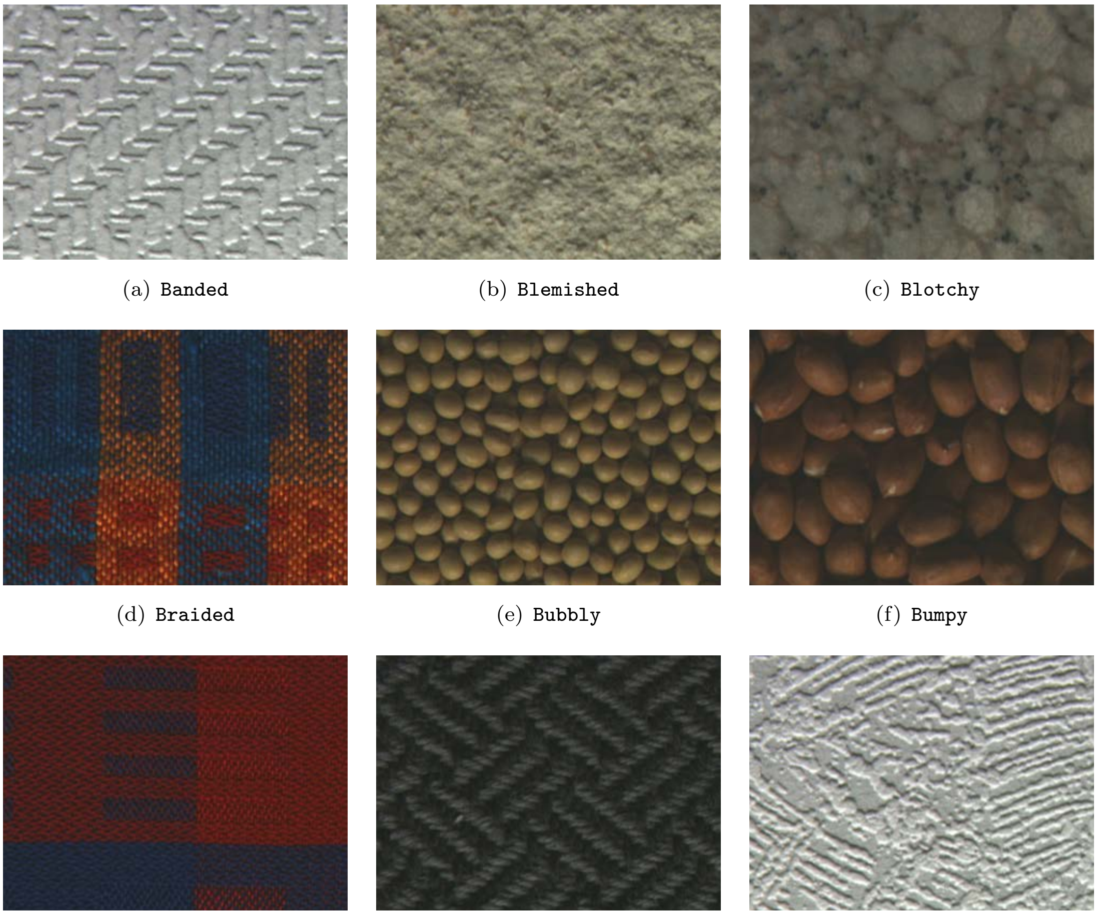
It was also demonstrated that there still exists a substantial semantic gap between numerous visual texture descriptors and the semantic space of texture. This is unimportant for most machine-centred applications of texture, but it is vital it be closed for human-focused tasks such as content-based image retrieval. Although it has long been known that certain categories of texture are more easily characterised by machines than others, it has been difficult to quantify the disparity and to guide further exploration. Our dataset, probabilistic framework, and semantic appraisal methodology are all tools that can be used to guide a principled exploration of novel texture features that align particularly closely with human perception. High-quality visual feature equivalents can be discovered on an attribute-by-attribute basis, and the semantic correspondence of those features can be readily evaluated using the appraisal experiment performed in this thesis.

We may also look to apply our results on a new dataset of visual texture that is more representative of the real-world, as is done in Cimpoi et al. (2014). In this paper the authors use a smaller version of the Bhushan dataset to label *wild* textures found through internet search engines, rather than captured under controlled conditions as done with the Outex dataset. This may be expected to better represent the kind of texture humans encounter in every day life, and that language has evolved to describe. They report excellent results in a retrieval experiment: in future work it would be very interesting to translate their approach to the pairwise comparison setting, and evaluate the benefits relative attributes bring to subsequent results.

Appendix A

Exemplar textures

The ninety-eight texture attributes of Bhushan et al. (1997) are shown in the figures below, along with an exemplar texture for each attribute calculated directly from our pairwise comparison dataset with the procedure of David (1987) (see Section 4.2.1).

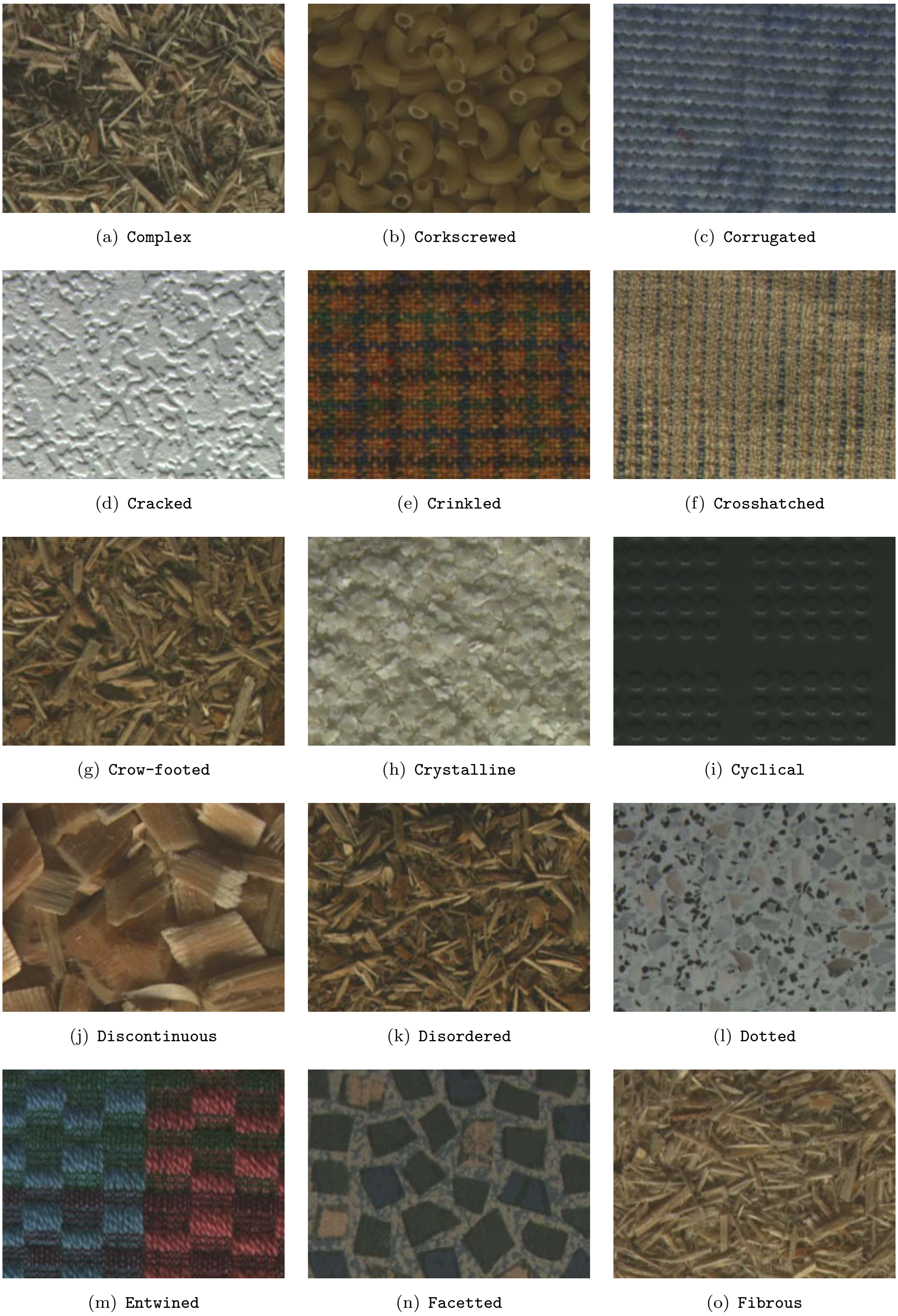


(a) `Banded` (b) `Blemished` (c) `Blotchy`

(d) `Braided` (e) `Bubbly` (f) `Bumpy`

(g) `Chequered` (h) `Cobwebbed` (i) `Coiled`

Figure A.1: Exemplar textures: `Banded` to `Coiled`.



(a) `Complex` (b) `Corkscrewed` (c) `Corrugated`

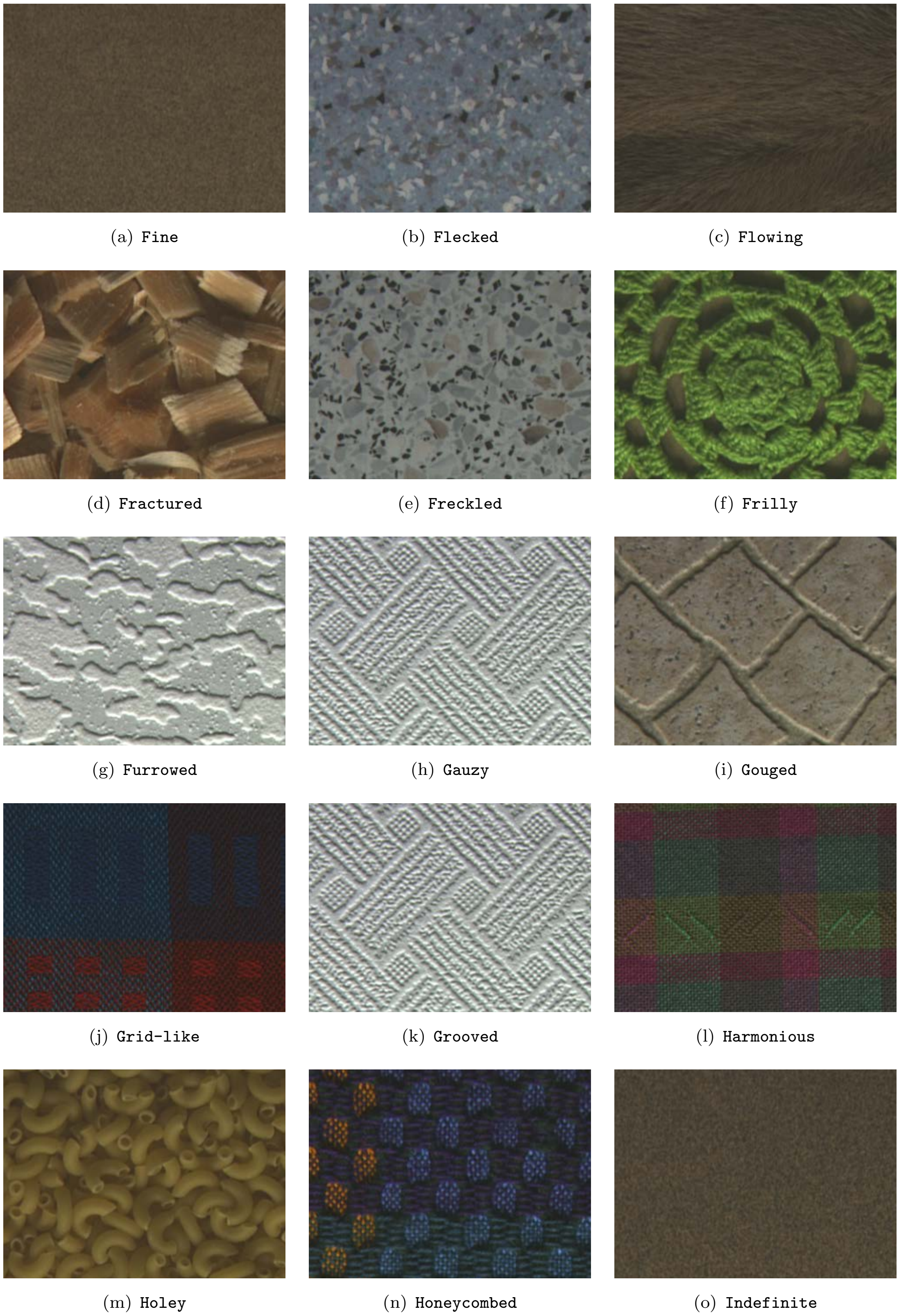
(d) `Cracked` (e) `Crinkled` (f) `Crosshatched`

(g) `Crow-footed` (h) `Crystalline` (i) `Cyclical`

(j) `Discontinuous` (k) `Disordered` (l) `Dotted`

(m) `Entwined` (n) `Facetted` (o) `Fibrous`

Figure A.2: Exemplar textures: `Complex` to `Fibrous`.



(a) `Fine`

(b) `Flecked`

(c) `Flowing`

(d) `Fractured`

(e) `Freckled`

(f) `Frilly`

(g) `Furrowed`

(h) `Gauzy`

(i) `Gouged`

(j) `Grid-like`

(k) `Grooved`

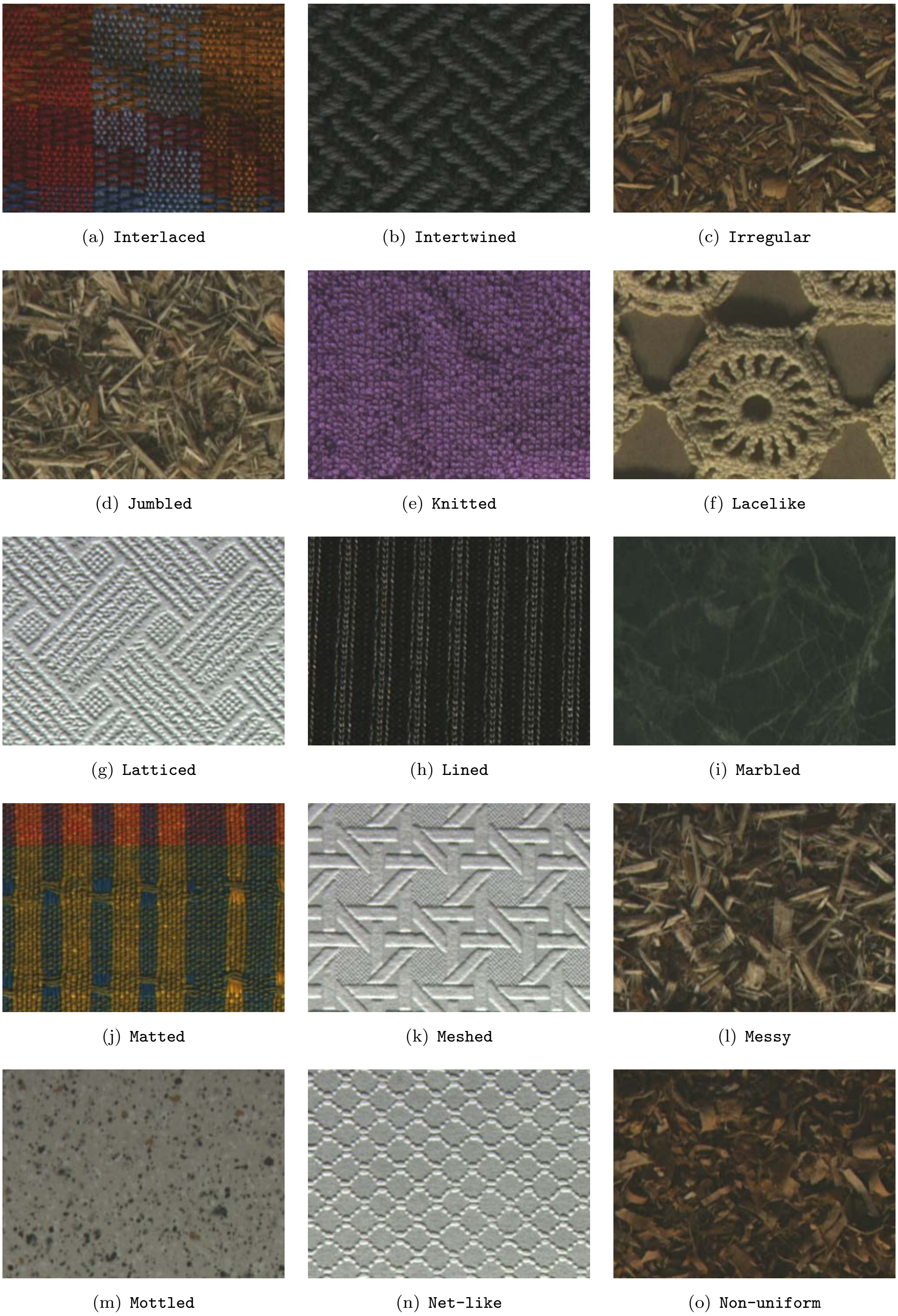
(l) `Harmonious`

(m) `Holey`

(n) `Honeycombed`

(o) `Indefinite`

FIGURE A.3: Exemplar textures: `Fine` to `Indefinite`.



(a) `Interlaced`

(b) `Intertwined`

(c) `Irregular`

(d) `Jumbled`

(e) `Knitted`

(f) `Lacelike`

(g) `Latticed`

(h) `Lined`

(i) `Marbled`

(j) `Matted`

(k) `Meshed`

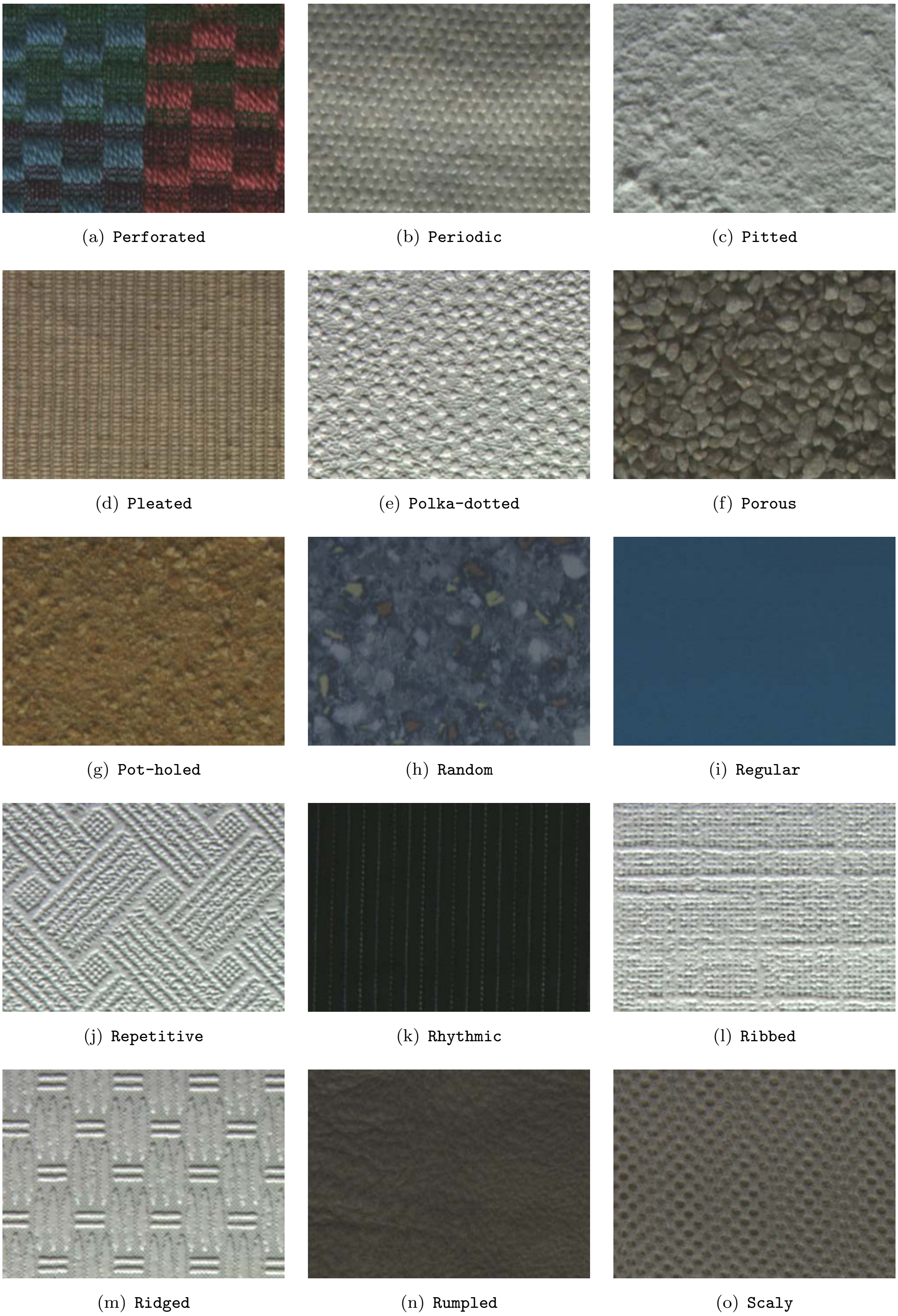
(l) `Messy`

(m) `Mottled`

(n) `Net-like`

(o) `Non-uniform`

FIGURE A.4: Exemplar textures: `Interlaced` to `Non-uniform`.



(a) `Perforated`

(b) `Periodic`

(c) `Pitted`

(d) `Pleated`

(e) `Polka-dotted`

(f) `Porous`

(g) `Pot-holed`

(h) `Random`

(i) `Regular`

(j) `Repetitive`

(k) `Rhythmic`

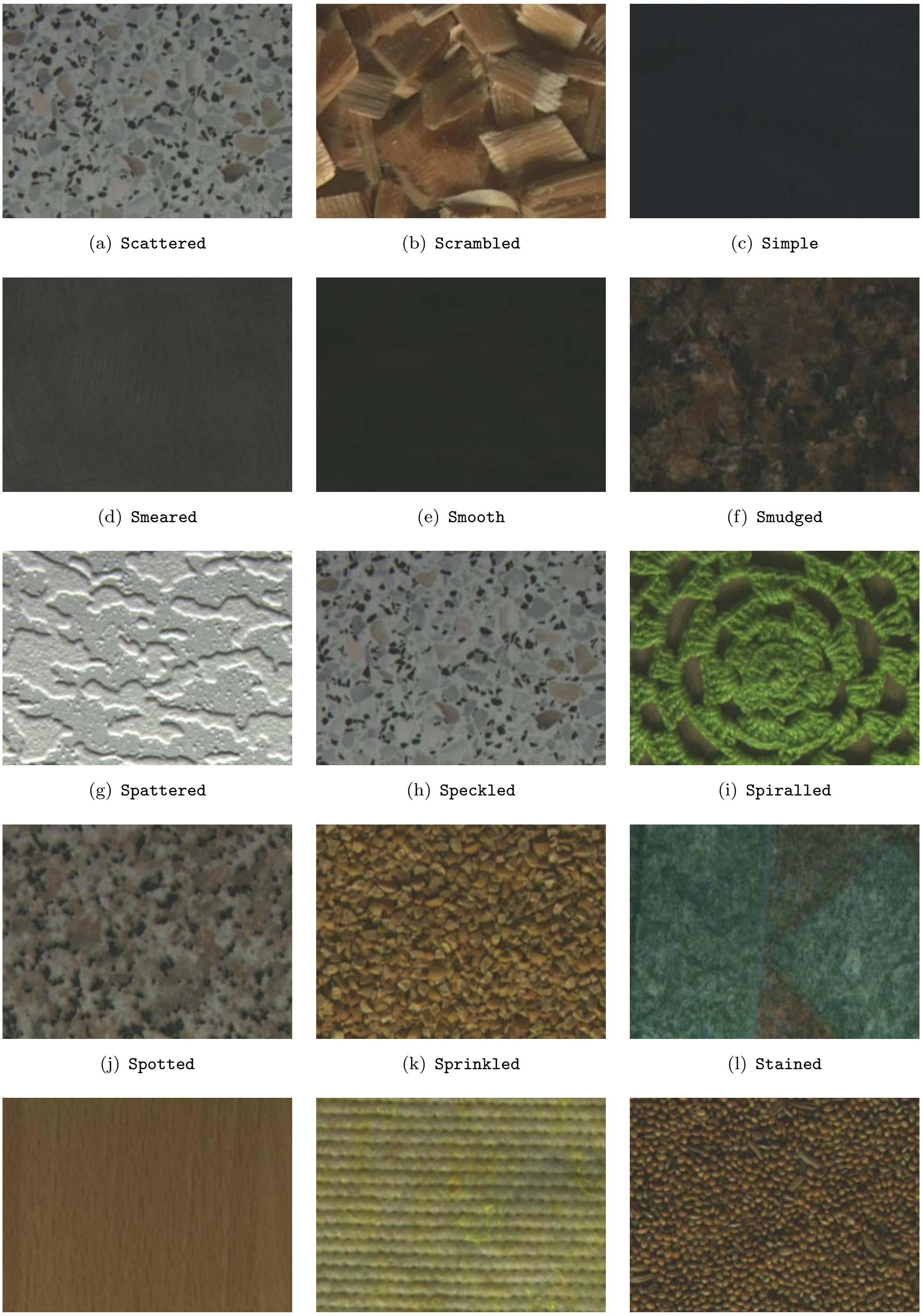
(l) `Ribbed`

(m) `Ridged`

(n) `Rumpled`

(o) `Scaly`

Figure A.5: Exemplar textures: `Perforated` to `Scaly`.



(a) `Scattered`

(b) `Scrambled`

(c) `Simple`

(d) `Smeared`

(e) `Smooth`

(f) `Smudged`

(g) `Spattered`

(h) `Speckled`

(i) `Spiralled`

(j) `Spotted`

(k) `Sprinkled`

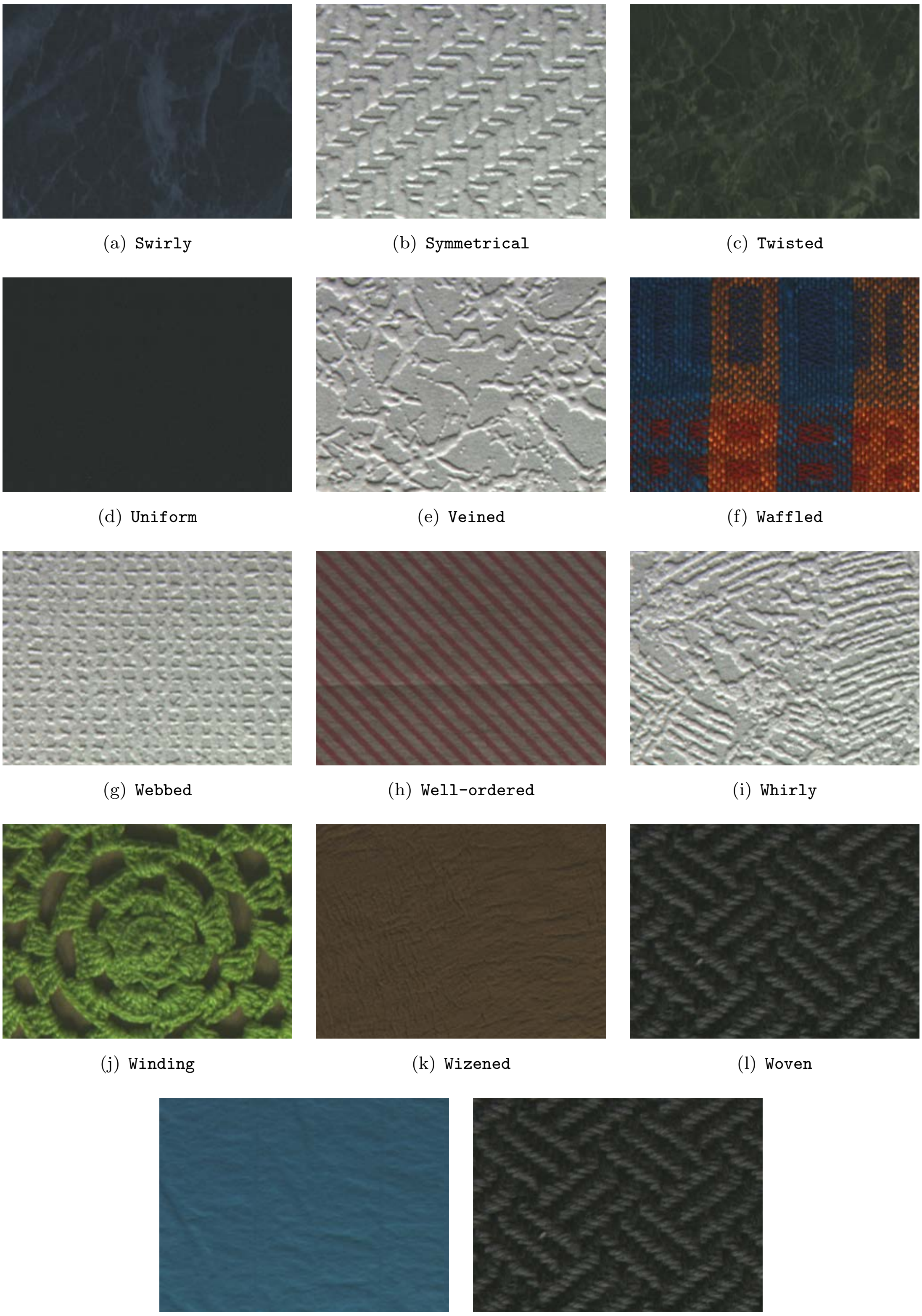
(l) `Stained`

(m) `Stratified`

(n) `Striated`

(o) `Studded`

FIGURE A.6: Exemplar textures: `Scattered` to `Studded`.



(a) `Swirly`

(b) `Symmetrical`

(c) `Twisted`

(d) `Uniform`

(e) `Veined`

(f) `Waffled`

(g) `Webbed`

(h) `Well-ordered`

(i) `Whirly`

(j) `Winding`

(k) `Wizened`

(l) `Woven`

(m) `Wrinkled`

(n) `Zigzagged`

Figure A.7: Exemplar textures: `Swirly` to `Zigzagged`.

Appendix B

Computational considerations with the Bayesian posterior

In Chapter 4 a Bayesian procedure for learning attribute strength ratings was introduced. Recall that the MAP estimate of the full $N \times P$ rating matrix $\mathbf{R}$ could be expressed:

$$\mathrm{vec}(\hat{\mathbf{R}}^{\mathrm{MAP}}) = (\mathbf{V}^{-1} \otimes \mathbf{U}^{-1} + \mathbf{C}^{\top}\mathbf{C})^{-1}(\mathrm{vec}(\mathbf{U}^{-1}\mathbf{M}\mathbf{V}^{-1}) + \mathbf{C}^{\top}\tilde{\boldsymbol{z}}) \tag{B.1}$$

The calculation of this estimate — and indeed, the calculation of many of the distributions associated with the Bayesian procedure (Equation 4.40, Equation 4.46, Equation 4.53) — involves a Kronecker product of two potentially large matrices and, resultantly, the inverse of a very large matrix. Below we discuss some of the means used to alleviate this complexity so as to achieve the results described in this thesis.

Firstly, it is often the case that a sparseness structure can be imposed on $\mathbf{V}$, $\mathbf{U}$, and $\mathbf{M}$ to simplify the time and memory requirements of the Kronecker product and subsequent calculations. If the between-attribute correlations are particularly weak then the single attribute multivariate normal formulation is usually sufficient instead of the full matrix-variate variety. Also, usually no specific prior information is known about the mean of $\mathbf{R}$ and so $\mathbf{M}$ can be set to zero, eradicating one part of the equation above.

Next, the block-diagonal structure of $\mathbf{C}$ admits various time-saving operations: the transpose and inverse of a block-diagonal matrix can be computed upon only its constituent blocks. The products $\mathbf{C}^{\top}\mathbf{C}$ and $\mathbf{C}^{\top}\tilde{\boldsymbol{z}}$ can also be computed in a blockwise fashion, so $\mathbf{C}$ need never be held in memory in its entirety.

Finally, the symmetric positive-definite $\mathbf{U}$ and $\mathbf{V}$ allow the equation to be solved iteratively using the *conjugate gradient method* (Hestenes and Stiefel, 1952). Here, particular systems of linear equations with a symmetric positive-definite coefficient matrix can be solved after k iterations, where k is the number of elements in the coefficient matrix, whilst avoiding the explicit calculation of that matrix. By expressing the equation above as:

$$(\mathbf{V}^{-1} \otimes \mathbf{U}^{-1} + \mathbf{C}^{\top}\mathbf{C})\text{vec}(\hat{\mathbf{R}}^{\text{MAP}}) = (\text{vec}(\mathbf{U}^{-1}\mathbf{M}\mathbf{V}^{-1}) + \mathbf{C}^{\top}\tilde{\boldsymbol{z}}) \tag{B.2}$$

we can see readily that $(\mathbf{V}^{-1} \otimes \mathbf{U}^{-1} + \mathbf{C}^{\top}\mathbf{C})$ is symmetric positive-definite due to the nature of $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{C}$.

Bibliography

G. I. Allen and R. Tibshirani. Transposable regularized covariance models with an application to missing data imputation. *The Annals of Applied Statistics*, 4(2):764–790, 2010.

D. M. Andrews and H. David. Nonparametric analysis of unbalanced paired-comparison or ranked data. *Journal of the American Statistical Association*, 85(412):1140–1146, 1990.

F. Arcizet, C. Jouffrais, and P. Girard. Natural textures classification in area V4 of the macaque monkey. *Experimental Brain Research*, 189(1):109–120, 2008.

M. Belkhatir. Combining visual semantics and texture characterizations for precision-oriented automatic image retrieval. In *Advances in Information Retrieval*, volume 3408 of *Lecture Notes in Computer Science*, pages 457–474. Springer, 2005.

B. Berlin and P. Kay. *Basic color terms: Their universality and evolution.* University of California Press, 1969.

N. Bhushan, A. R. Rao, and G. L. Lohse. The texture lexicon: Understanding the categorization of visual texture terms and their relationship to texture images. *Cognitive Science*, 21(2):219–246, 1997.

A. Budanitsky and G. Hirst. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources, Second Meeting.* North American Chapter of the Association for Computational Linguistics, 2001.

A. Cangelosi and S. Harnad. The adaptive advantage of symbolic theft over sensorimotor toil: Grounding language in perceptual categories. *Evolution of Communication*, 4 (1):117–142, 2001.

P. Carruthers. The cognitive functions of language. *Behavioral and Brain Sciences*, 25 (6):657–674, 2002.

D. Ceglarek, K. Haniewicz, and W. Rutkowski. Quality of semantic compression in classification. In *Computational Collective Intelligence. Technologies and Applications*, volume 6421 of *Lecture Notes in Computer Science*, pages 162–171. Springer, 2010.

O. Chapelle and S. S. Keerthi. Efficient algorithms for ranking with SVMs. *Information Retrieval*, 13(3):201–215, June 2010.

G. B. Chapman and E. J. Johnson. Incorporating the irrelevant: Anchors in judgments of belief and value. In T. Gilovich, D. Griffin, and D. Kahneman, editors, *Heuristics and biases: The psychology of intuitive judgment*, chapter 6, pages 120–138. Cambridge University Press, New York, 2002.

Y. Q. Chen, M. S. Nixon, and D. W. Thomas. Statistical geometrical features for texture classification. *Pattern Recognition*, 28(4):537–552, 1995.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613. IEEE, 2014.

S. Coradeschi and A. Saffiotti. An introduction to the anchoring problem. *Robotics and Autonomous Systems*, 43(2):85–96, 2003.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

H. A. David. Ranking from unbalanced paired-comparison data. *Biometrika*, 74(2): 432–436, 1987.

R. Desimone and S. J. Schein. Visual properties of neurons in area V4 of the macaque: sensitivity to stimulus form. *Journal of neurophysiology*, 57(3):835–868, 1987.

A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2352–2359. IEEE, 2010.

A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785. IEEE, 2009.

C. Fellbaum. *WordNet: An Electronic Lexical Database.* MIT Press, 1998.

Y. Fu, T. M. Hospedales, T. Xiang, J. Xiong, S. Gong, Y. Wang, and Y. Yao. Robust subjective visual property prediction from crowdsourced pairwise labels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):563–577, 2016.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis.* Chapman & Hall, London, 1995.

A. K. Gupta and D. K. Nagar. *Matrix variate distributions.* CRC Press, 1999.

A. Hanazawa and H. Komatsu. Influence of the direction of elemental luminance gradients on the responses of V4 cells to textured surfaces. *The Journal of Neuroscience*, 21(12):4490–4497, 2001.

R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.

S. Harnad. *Categorical Perception: The Groundwork of Cognition.* Cambridge University Press, 1987.

S. Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3): 335–346, 1990.

L. O. Harvey and M. J. Gervais. Internal representation of visual texture as the basis for the judgment of similarity. *Journal of Experimental Psychology: Human Perception and Performance*, 7(4):741–753, 1981.

M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6), 1952.

D. H. Hubel. *Eye, brain, and vision.* Scientific American Library, New York, 1995.

C. Hudelot, N. Maillot, and M. Thonnat. Symbol grounding for semantic image interpretation: From image data to semantics. In *Proceedings of the Workshop on Semantic Knowledge in Computer Vision, ICCV*. IEEE, 2005.

R. Jackendoff. How language helps us think. *Pragmatics & Cognition*, 4(1):1–34, 1996.

T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142. ACM, 2002.

J. P. Jones and L. A. Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6): 1233–1258, 1987.

B. Julesz. Visual pattern discrimination. *IRE Transactions on Information Theory*, 8 (2):84–92, 1962.

B. Julesz. Experiments in the visual perception of texture. *Scientific American*, 232: 34–43, 1975.

B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, 1981.

A. Karni and D. Sagi. Where practice makes perfect in texture discrimination: evidence for primary visual cortex plasticity. *Proceedings of the National Academy of Sciences*, 88(11):4966–4970, 1991.

F. A. Kingdom and D. R. Keeble. On the mechanism for scale invariance in orientation-defined textures. *Vision Research*, 39(8):1477–1489, 1999.

A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 453–456. ACM, 2008.

A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2973–2980. IEEE, 2012.

N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 365–372. IEEE, 2009.

V. A. Lamme, V. Rodriguez-Rodriguez, and H. Spekreijse. Separate processing dynamics for texture elements, boundaries and surfaces in primary visual cortex of the macaque monkey. *Cerebral Cortex*, 9(4):406–413, 1999.

C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE, 2009.

M. S. Landy and J. R. Bergen. Texture segregation and orientation gradient. *Vision Research*, 31(4):679–691, 1991.

K. I. Laws. *Textured image segmentation.* PhD thesis, University of Southern California, 1980.

D. Lewandowski, D. Kurowicka, and H. Joe. Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, 100(9): 1989–2001, 2009.

S.-S. Liu and M. Jernigan. Texture analysis and discrimination in additive noise. *Computer Vision, Graphics, and Image Processing*, 49(1):52–67, 1990.

Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, 2007.

J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America*, 7(5):923–932, 1990.

B. S. Manjunath and W.-Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.

B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, June 2001. ISSN 1051-8215.

D. Marr. *Vision: A computational approach.* Freeman & Co., San Francisco, 1982.

T. Matthews. *Crowdsourced texture comparison dataset.* University of Southampton, 2014. http://eprints.soton.ac.uk/id/eprint/397943.

T. Matthews, M. S. Nixon, and M. Niranjan. Enriching texture analysis with semantic data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1248–1255. IEEE, 2013.

W. H. Merigan. Cortical area V4 is critical for certain texture discriminations, but this effect is not dependent on attention. *Visual neuroscience*, 17(6):949–958, 2000.

A. Mojsilovic. A computational model for color naming and describing color composition of images. *IEEE Transactions on Image Processing*, 14(5):690–699, 2005.

A. Neviarouskaya, H. Prendinger, and M. Ishizuka. Sentiful: Generating a reliable lexicon for sentiment analysis. In *IEEE Transactions on Affective Computing*, pages 22–36. IEEE, 2011.

M. S. Nixon and A. S. Aguado. Introduction to texture description, segmentation, and classification. In *Feature Extraction and Image Processing*, chapter 8. Academic Press Inc., 3rd edition, 2012.

T. Ojala, T. Mäenpää, M. Pietikainen, J. Viertola, J. Kyllönen, and S. Huovinen. Outex-new framework for empirical evaluation of texture analysis algorithms. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 1, pages 701–706. IEEE, 2002a.

T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002b.

G. Okazawa, S. Tajima, and H. Komatsu. Image statistics underlying natural texture selectivity of neurons in macaque V4. *Proceedings of the National Academy of Sciences*, 112(4):351–360, 2015.

A. Oliva, A. Torralba, A. Guérin-Dugué, and J. Hérault. Global semantic classification of scenes using power spectrum templates. In *Proceedings of the International Conference on Challenge of Image Retrieval*, Electronic Workshops in Computing series. British Computer Society, 1999.

M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems*, volume 22, pages 1410–1418, 2009.

D. Parikh and K. Grauman. Relative attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 503–510. IEEE, 2011.

R. Penrose. On best approximate solutions of linear matrix equations. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 52, pages 17–19. Cambridge University Press, 1956.

J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70, 2000.

R. Rao and G. L. Lohse. Towards a texture naming system: identifying relevant dimensions of texture. In *Proceedings of the IEEE Conference on Visualization*, pages 220–227. IEEE, 1993.

D. Reid, M. S. Nixon, and S. V. Stevenage. Soft biometrics; human identification using comparative descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1216–1228, 2014.

D. A. Reid and M. S. Nixon. Using comparative human descriptions for soft biometrics. In *Proceedings of the International Joint Conference on Biometrics*, pages 1–6. IEEE, 2011.

B. E. Rogowitz, T. Frese, J. R. Smith, C. A. Bouman, and E. B. Kalin. Perceptual image similarity experiments. In *Proceedings of the SPIE Conference on Human Vision and Electronic Imaging*, pages 576–590. International Society for Optics and Photonics, 1998.

E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision*, volume 3951 of *Lecture Notes in Computer Science*, pages 430–443. Springer, 2006.

Y. Saad. *Iterative methods for sparse linear systems.* Siam, 2003.

S. Samangooei, B. Guo, and M. S. Nixon. The use of semantic human description as a soft biometric. In *Proceedings of the 2nd IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 1–7. IEEE, 2008.

P. H. Schiller. The effects of V4 and middle temporal (MT) area lesions on visual performance in the rhesus monkey. *Visual Neuroscience*, 10(04):717–746, 1993.

N. Serrano, A. E. Savakis, and J. Luo. Improved scene classification using efficient low-level features and semantic cues. *Pattern Recognition*, 37(9):1773–1784, 2004.

B. Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison, 2009.

E. Simpson, S. Reece, A. Penta, and S. Ramchurn. Using a Bayesian model to combine LDA features with crowdsourced responses. In *Online proceedings of the 21st Text REtrieval Conference*. NIST, 2013.

H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6):460–473, 1978.

S. Tominaga. A colour-naming method for computer color vision. In *Proceedings of the IEEE International Conference on Cybernetics and Society*, pages 573–577, 1985.

M. Tuceryan and A. K. Jain. Texture analysis. *The Handbook of Pattern Recognition and Computer Vision*, 2:207–248, 1998.

A. Tversky and D. Kahneman. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131, 1974.

C. Van Damme, M. Hepp, and K. Siorpaes. Folksontology: An integrated approach for turning folksonomies into ontologies. *Bridging the Gap between Semantic Web and Web*, 2(2):57–70, 2007.

J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision*, 72(2):133–157, 2007.

P. Vogt. The physical symbol grounding problem. *Cognitive Systems Research*, 3(3): 429–457, 2002.

G. Wang, D. Forsyth, and D. Hoiem. Comparative object similarity for improved recognition with few or no examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3525–3532. IEEE, 2010.

J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems*, volume 22, pages 2035–2043, 2009.

J. Zhang and T. Tan. Brief review of invariant texture analysis methods. *Pattern Recognition*, 35(3):735–747, 2002.

S.-C. Zhu, C.-E. Guo, Y. Wang, and Z. Xu. What are textons? *International Journal of Computer Vision*, 62(1):121–143, 2005.