Supervised Material Classification in Oblique Aerial Imagery Using Gabor Filter Features

by

Michael L Harris

B.S. Rochester Institute of Technology 2008

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Chester F. Carlson Center for Imaging Science
Rochester Institute of Technology

October 5th 2014

Signature of the Author _____

Accepted by _____
Coordinator, M.S. Degree Program Date

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE
COLLEGE OF SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Michael L Harris
has been examined and approved by the
thesis committee as satisfactory for the
thesis required for the
M.S. degree in Imaging Science

David Messinger Ph.D, Thesis Advisor

Harvey Rhody Ph.D

Carl Salvaggio Ph.D

Date

Supervised Material Classification in Oblique Aerial Imagery Using Gabor Filter Features

by

Michael L Harris

Submitted to the
Chester F. Carlson Center for Imaging Science
in partial fulfillment of the requirements
for the Master of Science Degree
at the Rochester Institute of Technology

Abstract

RIT's Digital Imaging and Remote Sensing Image Generation (DIRSIG) tool allows modeling of real world scenes to create synthetic imagery for sensor design and analysis, trade studies, algorithm validation, and training image analysts. To increase model construction speed, and the diversity and size of synthetic scenes which can be generated it is desirable to automatically segment real world imagery into different material types and import a material classmap into DIRSIG. This work contributes a methodology based on standard texture recognition techniques to supervised classification of material types in oblique aerial imagery. Oblique imagery provides many challenges for texture recognition due to illumination changes with view angle, projective distortions, occlusions and self shadowing. It is shown that features derived from a set of rotationally invariant bandpass filters fused with color channel information can provide supervised classification accuracies up to 70% with minimal training data.

Acknowledgements

This work would not be possible without the support and insight of my advisor Dr. David Messinger and the RIT Digital Imaging and Remote Sensing Lab and staff. I would also like to acknowledge my committee members for their guidance and instruction. Finally, I am incredibly appreciative of my officemates who, amid laughter and shenanigans, provided an enjoyable and stimulating working environment.

Contents

Chapter 1

Introduction

Because of the expensive nature of performing field campaigns and testing imaging sensors in real world conditions, simulation of their design and operation is often performed in virtually rendered environments. Within these environments, an incident spectrum, atmospheric parameters, object geometry, and material properties are loaded, and an internal physics based engine calculates the path of radiation propagation to finally model the sensor reaching radiance. This allows simulation of final image quality of an arbitrarily complex scene, and is an aid to developing sensor specifications, classification algorithms and the like.

To generate synthetic scenes, it is necessary to manually develop the 3D object geometry, and hand label all of the material types on each facet. This is very time consuming and prohibits the fast generation of larger scenes, and therefore the diversity that can be simulated within a scene. It also makes it difficult to generate synthetic imagery of real world scenes for simulation, since each object needs to be manually fabricated identically to the real world version. Given the constraints that manual model generation imposes, it is desirable to automate the procedure. This requires the automatic generation of both the geometry, and the material labels of a scene to be rendered. For the purpose of generating synthetic images of real world scenes this will involve the collection of data over a particular site of interest, transforming this data into both a geometrical model of the objects therein, and labeling each homogeneous object part with a material type. These can then be used as inputs to a synthetic image generation (SIG) software to simulate real world scenes.

Recent advances in computer vision that will be briefly discussed allow geometric

models to be generated from a collection of overlapping images in an automated workflow. In particular, this means that scene geometry can be created from aerial imagery, satisfying the need for one input in SIG software. The remainder of this work will focus on satisfying the second SIG input, the classification of the scene to be rendered into various material types. This will be done using the standard methodology of supervised classification. A feature set based on filter responses will be investigated, as well as the inclusion of color channel information. The emphasis is on determining how much training data (and therefore user interaction) is required, and whether standard texture features lead to adequate separability and classification accuracy on a challenging dataset of oblique aerial imagery taken over the Purdue University campus.

It is the goal of this work to develop and test a classification scheme that produces a salient class map that can be incorporated into SIG software to aid in automatic synthetic scene generation. The success of the classifier will be based on its overall accuracy, the individual class accuracies, its ability to incorporate diverse training data, and finally the quantity of training data necessary to produce acceptable results. Such a non-critical application does not require a high degree of accuracy or speed, but looking toward minimizing human intervention, the goal is minimal training data selected, automated parameter determination (if necessary), and high accuracy. The next sections in this chapter will briefly overview the operation and requirements of the SIG software at RIT called DIRSIG, the state of RIT's 3D extraction workflow, as well as detail the challenges of the aerial imagery used in the classification procedure developed. Finally, previous work which motivates the use of texture and color features in a supervised classification system will be discussed.

1.1 DIRSIG Scene Generation and Requirements

Modeling radiative transfer mechanisms along with the world's geometry allows simulation of the sensor-reaching radiance of a fabricated scene. RIT's Digital Imaging and Remote Sensing Image Generation (DIRSIG) tool provides an environment to model source geometry, atmospheric propagation, wavelength and geometry dependent spectral characteristics, and sensor operation [1]. DIRSIG can be used to complement real ground truth in, for example, the exploration of target detection or classification algorithm performance, and to simulate the effects of sensor design specifications on image quality. This section

Figure 1.1: DIRSIG Megascene bird's eye and detail views.

will provide an overview of DIRSIG's features and look into the user inputs required for generating synthetic imagery.

In addition to the source geometry, the required model inputs include the Bidirectional Reflectance Distribution Function (BRDF) within each representative material, as well as a texture map that facilitates spatial variation in the reflectance spectra. The material library has been generated from extensive ground truth measurements and contains hundreds of materials characterized under various conditions. This is used as a lookup table to induce the spectral character of each facet in the model.

Currently, base models of several types of vehicle, house, tree, industrial building and the like are fabricated by hand using a variety of Computer Aided Design (CAD) tools, and each material is specified. These base models are then attributed with varying materials and orientations and instantiated manually throughout the scene. Though scenes on the order of several square km have been generated, it is cost prohibitive to do this often or

dynamically for real world scenes. Automating this procedure would allow fast replication of real-world environments and structures, greatly increasing the diversity, complexity, and efficacy of simulation. This task requires the automatic generation of two inputs, namely the source geometry of the scene to be rendered, and a classmap of material type coupled to each facet of the geometry to leverage the DIRSIG material property database. Both of these inputs can be derived from images of the desired scene to be rendered.

Figure 1.1 shows a portion of a synthetic image constructed and rendered in DIRSIG that covers several square kilometers. To render this scene several external inputs are required. First the scene geometry must be in place. Figure 1.2 shows objects manually crafted from in-house and commercial CAD modeling packages. For each object several base models are created having different geometrical properties. Multiple building types, vehicles, and species of vegetation have been created as base models. To account for variability in color, texture, orientation, and size the base models are attributed with these additional properties and hand placed over the scene. For example a base model house may be fabricated then attributed with different roofing or siding types and subjected to several affine transformations to create a variety of houses in a residential scene. This leads to the ability to simulate diverse environments but is very time consuming and requires fabrication of a new base model any time a unique structure is simulated.

To employ radiometrically correct physical modeling of light interaction each facet of every instantiated object is labeled with a material linked to a database of material properties. The strategy for introducing objects with diverse spatial and spectral character is to draw from a large database of reflectance curves which contain variability induced by surface inhomogeneities as well as a BRDF model to account for variation in surface orientation. Spectral reflectance curves for a large number of materials have been built up over multiple large scale field campaigns to support simulation activity. In addition to reflectance every material has associated thermal properties such as emissivity allowing simulation in the infrared portion of the EM spectrum. Thus every facet requires a material label in order to leverage this library.

For rendering to accurately model sensor reaching radiance a robust atmospheric model also needs to be in place to account for scattering and absorption of EM radiation. DIRSIG can work with commonly used MODTRAN atmospheric profiles which specify solar illumination conditions and atmospheric conditions such as aerosol count, water vapor, and other molecular constituents. Finally, a sensor model is incorporated to include the effects

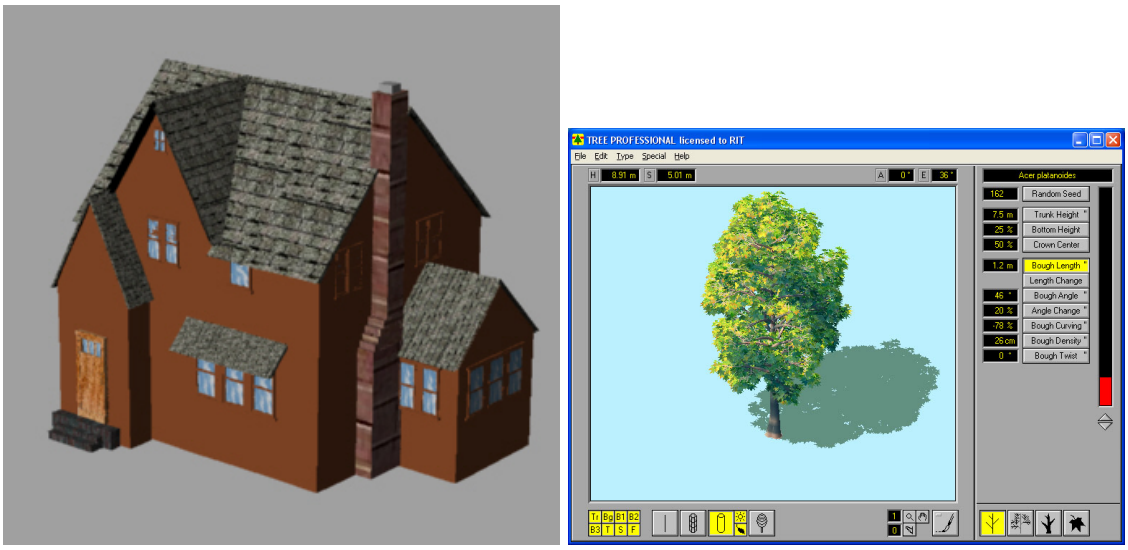


Figure 1.2: Object models generated by external software packages for inclusion in a DIRSIG simulated environment.

of various noise sources, integration over relevant bands, and spatial resolution on the output image.

Note that every input has spectrally dependent characteristics, leading to the ability to simulate a variety of imaging modalities. Figure 1.3 shows synthetic images of an aircraft generated utilizing visible and infrared bands in multi and hyperspectral configurations, as well as thermal and polarimetric bands. This reinforces the concept that for radiometrically accurate modeling each object facet requires specification of a particular material type and associated properties. The following sections will detail progress toward automating the construction of DIRSIG scene models particularly the generation of scene geometry, and material classification.

1.1.1 3D Extraction Workflow/Geometry Input

Toward automating construction of 3D models RIT has developed a workflow to automate point cloud extraction from a series of images based on point correspondences and photogrammetric principles [2]. Recently, the maturation of this work has allowed production of geo-located structural models [3]. These algorithms make it possible to capture aerial imagery of a scene from flyovers or passive remote monitoring, and quickly output a struc-

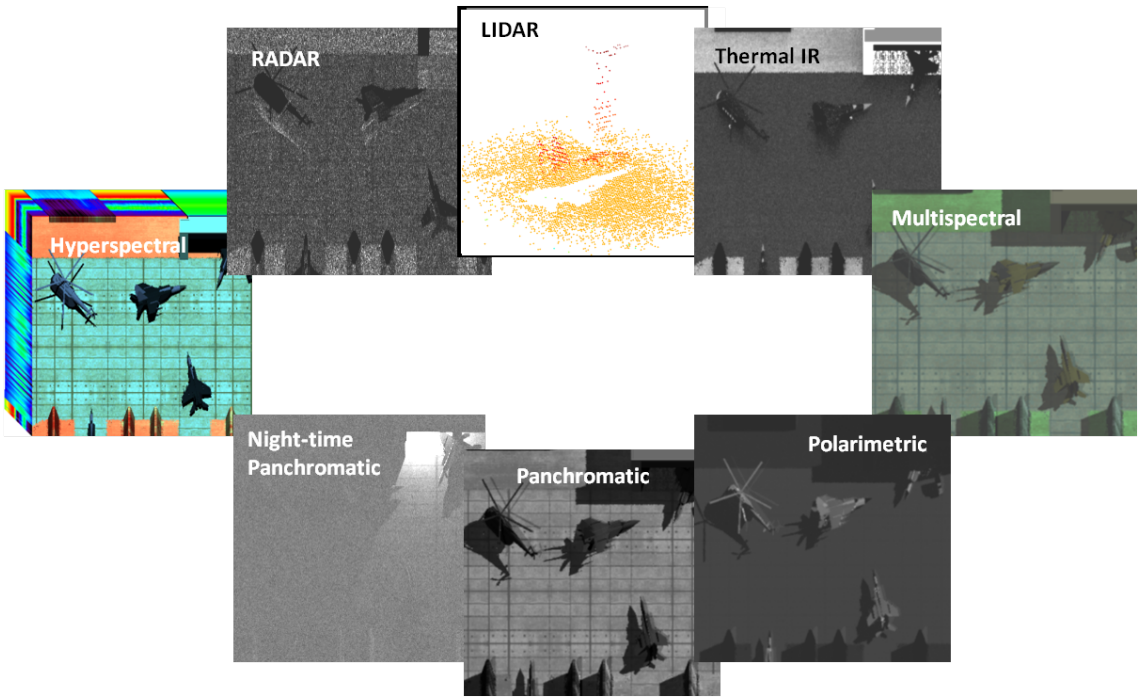
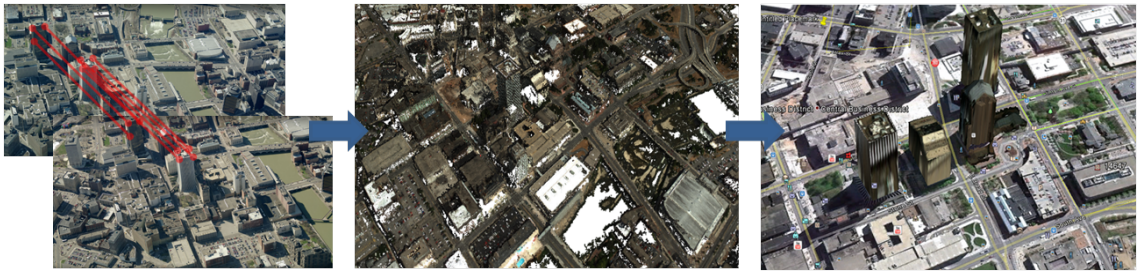


Figure 1.3: Synthetic scenes rendered in different wavelength ranges and modalities by DIRSIG.

(a)



(b)

Figure 1.4: 3D extraction workflow description (1.4a) and illustration (1.4b)

tural model useful for simulation or other tasks such as urban development or mission planning where knowledge of 3D scene geometry is critical. This represents a leap forward in automatic generation of source geometry for simulation.

Figure 1.4 shows a snapshot of the 3D extraction workflow and several facetized building models. As is evident the building sides do not contain much detail; this is due to the use of only nadir looking imagery. Research is ongoing to overcome the challenges associated with using oblique captures. Such challenges include finding correspondences among points which have undergone large rotations and perspective distortions and dealing with occlusions.

It is important to note how the availability of facetized models impacts the material classification procedure that is to be developed in the remainder of this work. The 3D extraction process results in images that are co-registered with a 3D model. In order to label a material type for each facet, some segmentation of the image must be done *a priori* that extracts the interesting components such as buildings at the object level. This allows the classifier to work on homogeneous regions such as rooftops and building façades rather than on the entire image. The work required to segment 3D models and label rooftops and façades is ongoing, and will be assumed to be possible for the purpose of this work. For the examples presented later, all segmentation is done manually, but in principle can

be done using the 3D data.

1.1.2 Material Label Input

The remainder of this work will address the second criteria of automated model generation for simulation: extraction of material type from images of the scene to be rendered. As discussed above this will be formulated as a supervised classification problem with the user identifying training regions in an input image of interesting material classes, and the classifier labeling all other regions in that and remaining images with a material type. As above, it is assumed that the sample images are already segmented into rooftops, building façades, and other object components (in this work this will be done manually). Next, the test dataset details are presented, and these and previous work will be used to select specifics of the classifier.

1.2 Data

The previous section laid out the problem of classifying material type from aerial imagery. This section will detail the specifics of the dataset that is available to test the procedure.

1.2.1 Specifics of this problem with oblique aerial imagery

Section 1.1 reviewed the input requirements for simulation of the image formation process of a modeled scene. These included both model geometry, and material labels for every facet of the model to render spatial/spectral interaction properties. In order to acquire accurate and geometrically detailed models, oblique views of the input structures are required. This also facilitates classification of building rooftops and facades. Aerial imagery with cameras poised at oblique angles provides a fast and reliable way of obtaining detailed imagery of large areas. Particularly interesting areas for simulation include urban and residential environments. As noted in the above considerations there are several challenges to classifying oblique aerial imagery of densely populated areas. These include the projective distortions associated with oblique look angles, varying solar illumination, varying object and texture scale, and the presence of occlusions and clutter.

Details of the data

The goal in acquiring a dataset was to develop and demonstrate a feature extraction and classification mechanism to identify the material type from oblique imagery of urban areas. A material classmap would then be combined with the geometry derived from the 3D extraction process of Section 1.1.1 which is being developed in tandem by other researchers at RIT. Data that was readily available included high resolution oblique imagery of the Purdue University campus from Pictometry Inc. Pictometry provides imagery of 4 look angles (pointed ahead, behind, right, and left of flightpath) and an orthorectified nadir shot. Only the standard RGB bands are available, and the resolution is about $\frac{1}{2}$ ft per pixel dependent on flying altitude.

Three collects of this area were performed, each at increasing resolution down to 2 inches, shown in Figure 1.5. At the coarsest resolution only the aggregated effects of texture are visible rather than individual shingles or brick and mortar. Higher resolutions offer richer textural content and may provide guidelines for the imaging conditions necessary to accurately label materials on oblique objects using only 3 bands. The following are details of the challenges of classification of oblique aerial imagery.

Projective Transformations

In purely nadir looking capture conditions all objects in the image lie approximately in the ground plane and transformations include primarily similarities (rotations and translations). For a single image the sensor elevation is also approximately constant so the scale at which objects and textures present themselves is constant. These transformations preserve angles and edge lengths so the underlying structure of any visible texture will remain the same. This is not the case in oblique imagery, where projective distortions vary this underlying structure such that a periodic rectangular array such as a brick wall will no longer appear rectangular, and additionally will be periodic with a different frequency.

Projective distortions also change the scale at which a texture feature occurs throughout the image. All image objects and textures are no longer an equal distance from the sensor, so the resolution varies pixel to pixel. Effectively this means that the same texture is sampled at different rates in different parts of the image, and will change in appearance due to aliasing. These are challenges to any feature extraction mechanism.

(a) (b)

(c)

Figure 1.5: Example images from the Purdue dataset at (a-c) 6, 4, and 2 inch GSD.

Illumination

Depending on the sun-sensor angle an image will exhibit variations in specularities, shadowing, and occlusions. These will affect both the pixel intensity and spectral character. Noise may also dominate in areas of low light, changing the appearance of texture. The spectrum of solar illumination will also vary with changing atmospheric conditions and date. This will have an effect on the reflected spectra of objects in the scene and will especially make use of spectral content difficult.

Clutter

A challenge to geometry extraction as well as material identification is the presence of occlusions and clutter in the scene. Clutter will be defined as any part of the image that is irrelevant to the classification mechanism and can possibly introduce error. A prominent source of clutter in urban scenes are windows. Any feature generation mechanism that operates over a region will pick up window pixels along with the building pixels that are interesting to the classifier, and essentially add a nonuniform source of noise. Other sources of clutter and occlusion include cars and other objects on rooftops that both add noise and obscure the underlying interesting regions.

1.3 Previous Work

After having a look at the dataset that is available and its challenges, we next consider related work that has been done to advance the field of scene classification. Both of these will inform the selection of an appropriate classifier, and its details will be developed.

1.3.1 Classification Task

Before looking at previous work, the details of the classification task will briefly be presented as they are relevant to the choice of classifier. The classification task can be divided into two subtasks, the first of which is selecting features of the data that allow for separability of unique classes in the dataset. It is not necessary for these features to completely describe the data, only to represent the data in such a way as to promote class discrimination. For example a pixel's value in the green channel taken as a feature from a multispectral image would probably allow separation of vegetation versus urban classes

but would not be adequate for distinguishing between types of vegetation. There are many texture and color features that are often used, and it is important to select those that best fit for the given dataset.

The second task is the labeling of regions in the image. This work will only consider the case of supervised classification rather than clustering because we wish to label regions with a certain class value rather than agnostically group the data according to mutual similarity. This task uses input training examples to model the appearance of a representative class in feature space and then labels regions with a class value according to a similarity criterion. Both of these tasks have been well studied and there is a wealth of related work dealing with optimal feature generation, and advanced classification algorithms. The following will detail related work in the area of feature selection, and classification.

Feature Generation Task

The first task of a classifier is to represent an image in terms of features that lead to optimal separation of the constituent classes in feature space. Because in this work we are dealing with three band RGB imagery it seems likely that textural metrics will provide necessary features in addition to the color channel intensities. For the purpose of material identification, textural metrics seem necessary in addition to color information due to aesthetic variations in materials such as brick, siding, and residential roofing. Therefore we are also interested in determining the extent to which color and texture features can or should be combined to enhance classification.

Previous work in texture analysis for the purpose of classification has focused on four main texture metrics, these being statistically derived (such as moments), model based, structural and geometrical, and finally filter based [4]. No single feature set has been discovered which should be used in every case, rather constraints on computational complexity, ease of implementation, and efficacy for a given dataset should guide the choice of feature set. Much recent work has been dedicated to the understanding and advancement of filter based methods [5, 6, 7, 8, 9] and excellent classification results have been achieved on a wide variety of texture databases, natural imagery, and moderate resolution satellite imagery. The major advantage of filter based approaches is they are inherently multiresolutional [10]. Traditional methods such as gray level co-occurrence features [11] analyze moments over the entire image spectrum, whereas a multiresolution approach is able to parse the frequency domain of an image and analyze localized regions in the spectrum.

This leads to the potential for increased discrimination of textures of varying frequency content. The review paper of Randen [6] concludes extensive tests with the result that, although there is no clear winning feature extraction technique for every case, filter based methods typically outperform the traditional methods. In addition, the GLCM method is more computationally demanding than filter based methods. As pointed out in [5] the Gabor filter implementation in [6] could have been improved with better filter spacing.

Each of the above philosophies of feature extraction focuses on representing texture in various ways, and pulls out features that should be indicative of different textured regions. The mean, variance, and other higher order moments are examples of statistical descriptors that attempt to quantify and discriminate between texture regions. The filter based approaches produce features based on the response of a given texture region to a bank of filters, and can be quickly applied in the frequency domain. It is not the purpose of this work to develop a novel feature generation technique, but rather to investigate the application of standard techniques to an interesting dataset. Therefore, because of the fundamental underpinning of the filter based methods in frequency analysis, we will focus on these methods.

Classification Task

Once salient features have been identified and extracted the supervised classifier's task is to group the data according to similarity to the training samples. Most classifiers can be thought of as methodologies of constructing decision boundaries around the training data, such that a point in feature space which falls within a given boundary, is labeled a member of that particular class. Concerns similar to selecting a feature set exist in selection of a classifier, such as computational intensity, and whether the decision boundary shapes (hypersphere, hyperellipsoidal) fit the data.

1.3.2 Classification Based On Texture

Because the data available is three band imagery, the color content alone may not be enough to ensure good classification. Therefore we first look at previous work involving classification using texture features.

Gabor Filters

Because of similarity to the human visual system [12], the Gabor filter has garnered much attention for texture analysis [13, 14] in the areas of segmentation [10, 5], content-based image retrieval [15, 7], as well as supervised classification [16, 17], and facial recognition [18]. Typically, texture databases have been used to test various configurations of a Gabor filter bank on classification accuracy. In cases where remotely sensed data were used, the resolutions have been moderate, and the textures have consisted primarily of nadir looking satellite images of broad regions of built-up, forested, or water.

The Gabor filter has been favored over other filters [6] due to its optimal localization in the spatial and spatial frequency domains [19], and its flexibility to be arbitrarily specified in many configurations. It can be easily specified at any spatial frequency center frequency, orientation, and with any bandwidth. This makes it suitable as a bandpass filter for extracting dominant frequencies in textured images. It is also mostly insensitive to global variations in intensity, which is a necessary feature in classification of real scenes.

Good results have been achieved in supervised classification of satellite imagery [16, 17], however there have been no studies using this filter based approach on higher resolution aerial imagery for the purpose of material identification. Because of the promising previous results, and desirable properties, the Gabor filtering scheme will be tested on the novel dataset described above.

Invariant Texture Features

The nature of the test dataset ideally requires some invariance to affine distortions, rotations, and illumination variation, described in Section 1.2.1. Features based on texture rather than spectral character immediately provide some robustness to changing illumination levels, though less to changing illumination directions. Previous work also provides a simple way of providing some rotation invariance to the Gabor filtering scheme, which will be described in a later chapter. The Gabor filter's spatial extent and flexible control over its bandwidth should also provide some degree of invariance to other affine distortions, since these are similarly manifested in the frequency domain.

Classification

Once a feature set has been generated, in this case using texture features, the remaining task is to label image regions according to similarity to training data. The simplest way of doing this is with a minimum distance classifier, which assigns each region a class label based on its direct proximity in feature space. In this work we are mainly concerned with the efficacy of the feature set, so the simplest approach will be used. The Bhattacharyya distance is commonly used to compute separation between two distributions in feature space [20], and will be used here to measure distances between distributions taken from test image regions, and each training distribution. To measure distances from an individual feature vector to a training distribution the Mahalanobis distance will be used [20]. Both of these distances are normalized by the covariance of the training (and the test distribution in the former case) data and assume hyperellipsoidal class boundaries, which are suitable for use with Gabor filters [5]. Because both of the previous distances assume normal data, another minimum distance method will be computed by packaging the data as a histogram. This method avoids covariance estimations, and statistical assumptions, and is easy to apply as well.

1.3.3 Combining Color and Texture

Although the color channel information may prove to be erroneous due to the changing illumination conditions over multiple images, we will examine several ways of doing so and their impact on the resulting classification accuracy.

Color Spaces

The standard RGB image representation is highly correlated from band to band, so for this work we will use the alternative $L^*a^*b^*$ color space described in [21]. This color space has the advantage that its bands are approximately orthogonal, and additionally, split the spectrum into an overall illumination or intensity component, and two chromaticity components. Thus, for texture analysis, the intensity channel can be filtered, and the chromaticity components can be used for spectral analysis.

Color Texture Fusion

Previous work has shown several methods of combining color and texture information [22, 23, 24, 25]. In [22] Gabor filtering of each color channel was proven a successful method of extracting color-texture features. The major lines of research have investigated joint and separate color-texture processing. One naïve method, and two other simple methods which have shown promise will be investigated.

The first is to first generate texture features, then simply concatenate the color channel intensities onto each texture feature vector. The next is to filter every color channel, generating a texture feature vector for each channel, and concatenating those, as in [23]. The last is to classify using only the texture features, and again using only the color channel intensities, then combine the outputs of the two classifiers to generate the final classmap. Work on multiple classifier fusion such as [26] provides very simple voting methods that will be used here.

1.3.4 Conclusion

The problem of classifying oblique aerial imagery for the purpose of synthetic image generation has been presented along with the challenges of this particular dataset. The methodology for classification takes advantage of the fact that the 3D processing leads to a geometric model that is co-registered with the images, such that whole buildings, as well as their rooftops, facades, and other components can be segmented out of the images, leaving ideally homogeneous textured regions. Because of the deficiency in color information, the texture information should provide additional discriminatory power for producing a thematic map of material classes. Although there are many textural metrics that have been generated, because of flexibility, widespread use, ease of implementation in the frequency domain, robustness to affine and illumination variations, and proven success in many areas, the Gabor filtering scheme will be used to extract textural information. Several methods of fusing color channel information will also be used.

The main goal of the following sections is to determine the efficacy of commonly used textural features in classifying a dataset which presents novel challenges, as well as the extent to which color channel information can or should be used in the task. Because the dataset includes three different resolutions, the results assessment should give some guidance as to the resolution required to adequately capture textural information with discriminatory power. Other logistical challenges include the order of operations of the

classification task - should test regions first be segmented and the textural features aggregated before classification, or should each pixel be classified and then the classmap aggregated? Since there is no clear answer, both methods will be investigated.

Chapter 2

Methodology

2.1 Filtering for Texture Analysis

This section will detail the application of a filter bank for texture analysis, the type of filters used and their suitability for this application, the nature of the derived feature sets, and the classification algorithm used.

2.1.1 Overview of Filtering Scheme

The filtering approach to texture analysis is grounded in the fact that textured images can be represented by their frequency content, or lack thereof, and by designing a filter which responds to the appropriate frequencies, textural metrics can be extracted. These metrics can then be used for concise texture representation, image reconstruction, and discrimination among various textures [16, 5, 27]. Figure 2.1 details a schematic of this process in the context of classification of textured images. First an input texture image, assumed to be homogeneous, and contain one or more dominant frequencies, is convolved with a filter tuned to one of those dominant frequencies. Next, a feature is generated by taking some non-linearity, such as an energy function, and smoothing is performed. Therefore a textured image is represented by the energy contained in its dominant frequency band.

Two or more textures can be segmented by estimating their dominant frequencies, and constructing one or more filters (equal to one less than the number of textures) tuned to these dominant components. In the case of many textures, or where it is inefficient to first estimate all of the dominant components, a filter bank which covers the entirety of the

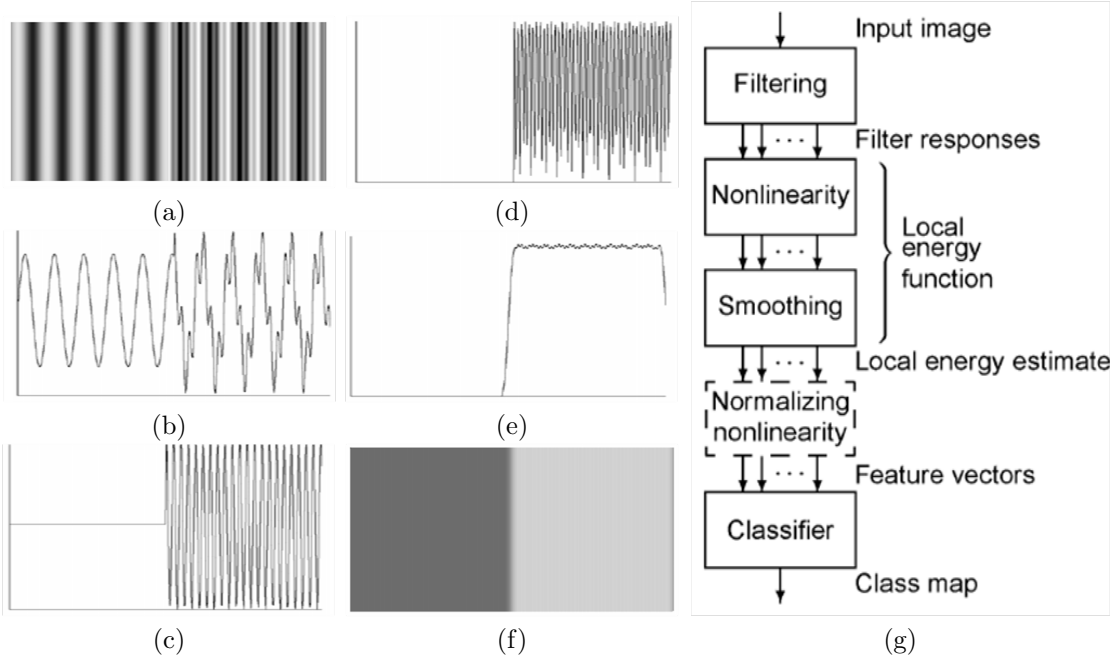


Figure 2.1: Process of applying a single filter to an input texture image of two dominant frequencies (a). An amplitude profile of the image (b). Result of applying filter tuned to the texture in the right half of input image (c). Rectification (or other non linearity) (d), and finally smoothing (e). The output amplitude contrasts the two different textures (f). The process in algorithmic form (g). Graphic included from [6]

frequency spectrum can be designed [10]. In this case each texture is then represented by a feature vector of responses to each filter, and classification is performed. The particular filter choice and placement in the frequency domain depends on the location of relevant frequencies in the image, and will be discussed below.

2.1.2 Gabor Filter

Gabor Wavelet

The methodology of decomposing a signal into its dominant frequency components has been well studied in signal and image processing. One special criteria for local texture recognition is localization of the filter in both space and spatial frequency. It has been shown that the (complex) Gabor filter satisfies this criteria in the sense that it minimizes the spatial/spatial frequency uncertainty relation [13, 19]. This is to say that it optimally

isolates spatial frequency content in a localized portion of the image [1].

The 2D Gabor wavelet is a Gaussian function modulated by a complex sinusoid as in

$$g(x, y) = s(x, y)w_r(x, y) \tag{2.1}$$

where the complex sinusoid is denoted by $s(x, y)$, the Gaussian envelope by $w_r(x, y)$, both with 2D spatial coordinates (*x,y*). The sinusoid has the effect of specifying the center frequency and orientation of the filter. This produces a spatial frequency bandpass filter that can be arbitrarily located and oriented.

The complex sinusoid is given by

$$s(x, y) = \exp(j(2\pi(u_0x + v_0y) + \Phi)) \tag{2.2}$$

with frequency location u_0 and v_0, and initial phase Φ. In polar coordinates the magnitude F_0 and direction ω_0 are respectively

$$F_0 = \sqrt{u_0 + v_0}, \quad \omega_0 = \tan(\frac{u_0}{v_0}) \tag{2.3}$$

and the equation becomes

$$s(x, y) = \exp(j(2\pi F_0(x\cos\omega_0 + y\sin\omega_0) + \Phi)). \tag{2.4}$$

The real and imaginary components of the complex sinusoid (Figure 2.2) are respectively

$$\mathrm{Re}(s(x, y)) = \cos(j(2\pi(u_0x + v_0y) + \Phi)) \tag{2.5}$$

$$\mathrm{Im}(s(x, y)) = \sin(j(2\pi(u_0x + v_0y) + \Phi)). \tag{2.6}$$

The Gaussian envelope (Figure 2.2) is specified as

$$w_r(x, y) = K\exp(-\pi(a^2(x - x_0)_r^2 + b^2(y - y_0)_r^2)) \tag{2.7}$$

[1] However there exist other optimality criterion—see [12, 28].

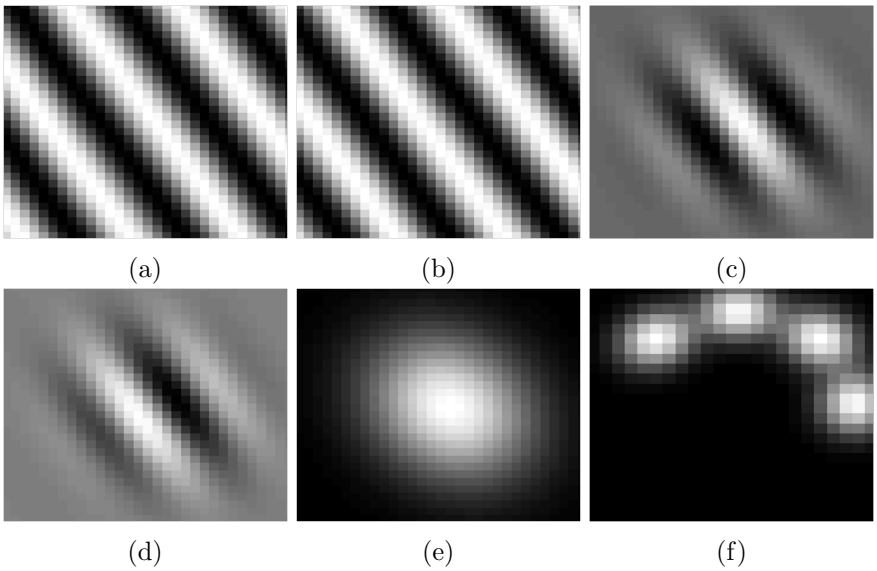


Figure 2.2: Real (2.2a) and imaginary (2.2b) parts of the complex sinusoid component of a single Gabor function. Real (2.2c) and imaginary (2.2d) parts of the Gabor function. Gaussian component (2.2e), and frequency domain showing 4 Gabors (2.2f) from 0 to 135 degree orientations.

with subscript r indicating a rotated coordinate system given by

$$(x - x_0)_r = (x - x_0)\cos\theta + (y - y_0)\sin\theta \tag{2.8}$$

$$(y - y_0)_r = -(x - x_0)\sin\theta + (y - y_0)\cos\theta, \tag{2.9}$$

and a and b give the extent of the ellipse along the major and minor axes. This windowing function produces the desirable spatial/spatial frequency localization properties of the Gabor function. It is also apparent that the real and imaginary components of the function are nearly equivalent but for a 90 degree shift in phase, so they approximately form[2] a quadrature pair. The analytic portion of any filtered signal can then be extracted by discarding the phase.

[2]The cosine component contains some DC response which must be eliminated to produce an actual quadrature pair.

Properties

We will now investigate the specification of the above parameters and their relation to the multichannel filtering approach to texture recognition. The Gabor function in polar coordinates is

$$\begin{aligned} & g(x,y) = \\ & K \exp(-\pi(a^2(x-x_0)_r^2 + b^2(y-y_0)_r^2)) \\ & \exp(j(2\pi F_0(x\cos\omega_0 + y\sin\omega_0) + \Phi)). \end{aligned} \tag{2.10}$$

This can be rewritten in terms of a standard deviation σ and aspect ratio λ as

$$\begin{aligned} & g(x,y) = \\ & \frac{1}{2\pi\lambda\sigma^2} \exp(-\frac{(\frac{1}{\lambda^2}(x-x_0)_r^2 + (y-y_0)_r^2)}{2\sigma^2}) \\ & \exp(j(2\pi F_0(x\cos\omega_0 + y\sin\omega_0) + \Phi)). \end{aligned} \tag{2.11}$$

where, for normalization to a maximum of unit magnitude in the frequency domain, we require $K = \frac{1}{2\pi\lambda\sigma^2} = ab$.

In the frequency domain, the Fourier transform of the Gabor function gives

$$\begin{aligned} & G(u,v) = \\ & \exp(-2\pi^2\sigma^2(\lambda^2(u-u_0)_r^2 + (v-v_0)_r^2)) \\ & \exp(j(-2\pi(x_0(u-u_0) + y_0(v-v_0) + \Phi))) \end{aligned} \tag{2.12}$$

with magnitude and phase in polar coordinates

$$\text{Magnitude(G(u,v))} = \exp(2\pi^2\sigma^2(\lambda^2(u-u_0)_r^2 + (v-v_0)_r^2)) \tag{2.13}$$

$$\text{Phase(G(u,v))} = \exp(j(-2\pi(x_0(u-u_0) + y_0(v-v_0) + \Phi))). \tag{2.14}$$

The form of this equation produces an ellipse with center located at (u_0, v_0) or (F_0, ω_0), major axis length proportional to σ and aspect ratio λ. This allows specification of a bandpass filter of arbitrary location and orientation. In general only filters with polar angle equal to orientation angle are useful since any other configuration will lead to an orientation dependent bandwidth. For this form the frequency and orientation bandwidths

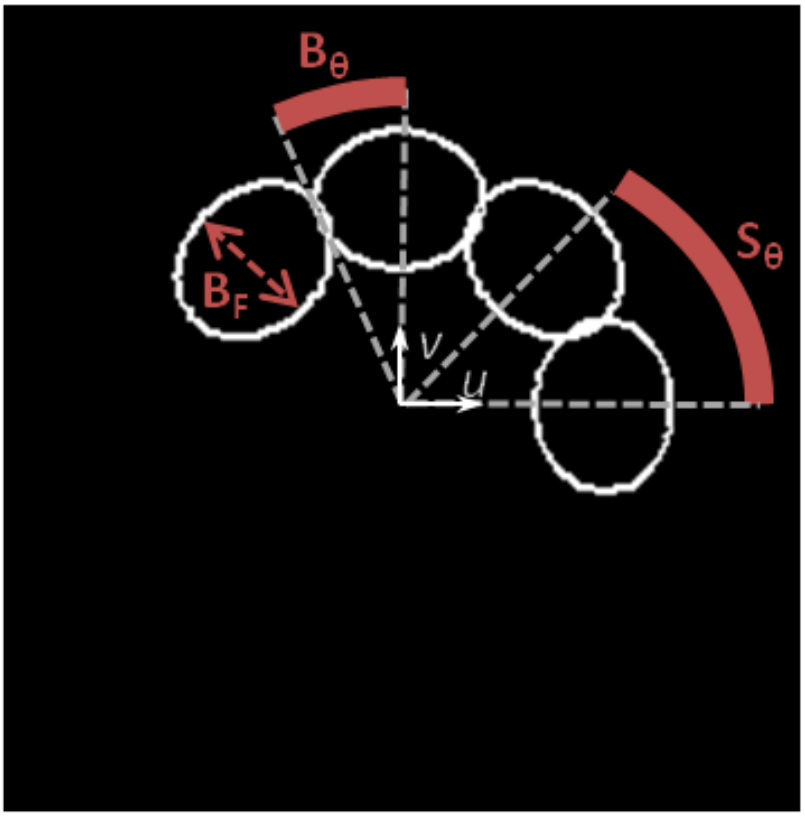


Figure 2.3: Specification of the frequency, B_F, and orientation, B_θ, bandwidths, and angular spacing S_θ of a Gabor wavelet.

can be derived [29] respectively as

$$B = \log_2\left(\frac{\pi F\lambda\sigma + \alpha}{\pi F\lambda\sigma - \alpha}\right) \tag{2.15}$$

where $\alpha = \sqrt{\log_2 2/2}$ and

$$\Omega = 2\tan^{-1}(\alpha/\pi F\sigma). \tag{2.16}$$

Figure 2.3 shows the details of these specifications on the frequency spectrum of a Gabor wavelet.

Thus it is possible to construct a bank of filters of varying center frequency and orientation to achieve complete coverage of the frequency domain. A textured image can then be represented by its responses to a set of spatial frequency filters that are only applied to

a localized image region whose size is dependent on the characteristic length σ and aspect ratio λ. This representation can be thought of as a localized expansion of the image using the Gabor wavelets as an independent set of basis functions. The energy contained in the coefficients of the expansion are then used as textural features.

Filter Bank Design

Since the frequency content of an image and its constituent classes may not be known *a priori*, it may be necessary to incorporate a bank of filters each tuned to a separate portion of the frequency domain. As above there are many free parameters of the Gabor filter that can be tuned, which can essentially be broken down into the frequency bandwidth, center frequency, and orientation. Furthermore for a bank of filters, the number of filters and their angular and frequency spacing is also variable. It is important to specify these parameters to maximize information extraction.

The dyadic configuration in [10] is widely used because it produces nearly uniform and entire coverage of the spatial frequency domain. Figure 2.4 shows examples of this arrangement. The foundation of this approach is in the assumption that most texture content is found at lower frequencies, and so the resolution should be higher there for increased discriminatory power. As a consequence of this arrangement, higher frequencies are sampled with lower resolution.

This leaves the angular spacing, the bandwidth, and the number of filters to be set. [5] uses an angular spacing of 30° , and in [10] the number of filters is set such that the highest frequency filter's center frequency lies just inside the window of Nyquist aliasing. Since the bandwidth is probably imagery specific, it will be left as a free parameter to be tuned on each imagery dataset used. Heuristic guidelines for its range appear in [16], to be between 0.7 and 1.5.

2.1.3 Feature Extraction

Filter Bank Feature Extraction

As shown in Figures 2.1 an input image is convolved with a bank of filters each tuned to a different portion of the frequency domain. The filtered output at this stage is still oscillatory in the spatial domain, and requires some form of rectification and averaging to produce uniform responses to each texture class. Several ways of performing the rec-

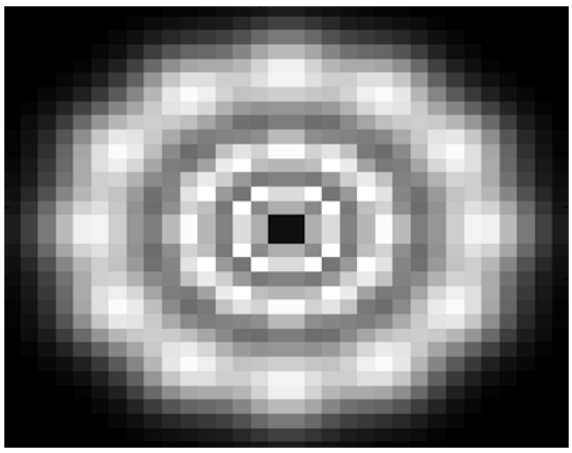


Figure 2.4: 32×32 Dyadically spaced Gabor filter bank containing 4 center frequencies and 12 orientations. Each filter center frequency is one octave above the lower one.

tification have been proposed and investigated, and all involve application of a non-linear function, most often one that computes the energy (complex magnitude) of the filtered response [5]. Others have included taking only the real component of the filtered output, absolute value, and use of a sigmoidal function.

Since this has been thoroughly investigated we will use the simple approach of taking the absolute magnitude of the filter responses to derive one feature per pixel of the convolved image, as well as the variance of the area within the filter extent to derive a second feature. These features were shown to be effective in [5]. Additionally using the simple absolute magnitude avoids additional parameters in defining a suitable non-linear function.

Following rectification, smoothing can be performed to produce a uniform output. To do this a smoothing function is applied to each rectified filtered output that is the same shape and orientation as the Gaussian component of the filter used to generate the output, but with a wider extent [16, 5]. The smoothing parameter γ is used to control the extent of smoothing simply by scaling the coordinate system of the Gaussian component as in

$$g(x, y) \rightarrow g(\gamma x, \gamma y). \tag{2.17}$$

The recommended value of γ in [5] is $\frac{2}{3}$.

Rotation Invariance

Several authors have provided an elegant method for deriving rotationally invariant features from the outputs of the filter responses detailed above [30, 31]. This method relies on the fact that along the orientation axis, the filter responses are periodic, and therefore a rotation of the image simply shifts the response along the rotation axis. As long as there is adequate coverage then the response of an image with textures oriented at 0° will ring the 0° oriented filter, and when rotated, will excite an adjacent filter tuned to that orientation. This means that rotations of the input image are mapped to translations along the orientation axis.

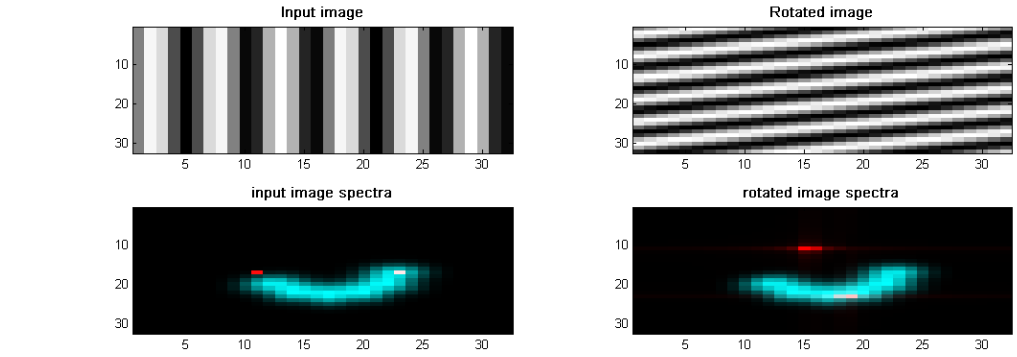
Because of this periodicity and mapping, if a DFT is taken along the orientation axis for each response, the property of shift invariance will lead to identical DFT coefficients for images undergoing only rotational transformations. Half of the DFT coefficients are redundant due to even symmetry when operating on real signals but the remaining coefficients are rotation invariant. This procedure can be used to achieve rotation invariance on both the rectified signal outputs, and the variance outputs. Figure 2.5 shows this procedure by filtering two rotated sinusoids. The rotated spectra fall on different Gabor wavelets leading to a shift in the overall response along the orientation axis. The response of an individual pixel in the 45° sinusoid is seen to shift two indices to the right in the plot of Gabor coefficients, but the 45° and 0° sinusoids DFT coefficients line up well.

2.2 Classification

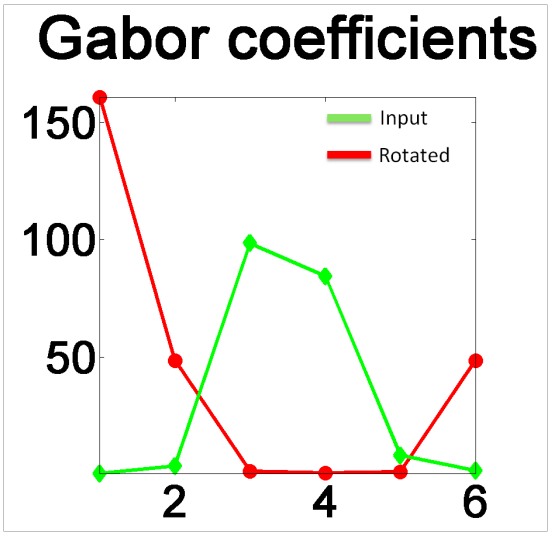
After deriving features which in this work are textural metrics derived from the output of filter responses each region is assigned a class label using a distance metric. In this work a simple minimum distance classifier is used rather than a more advanced and computationally expensive one so that the emphasis is on the choice of feature set, rather than the choice of classifier.

2.2.1 Minimum Distance

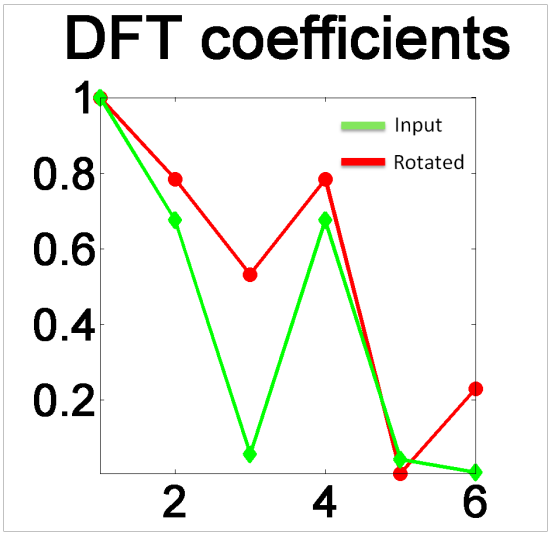
The minimum distance classifier labels a particular region with the class label of the overall class distribution it is closest to in feature space. Two minimum distance classifiers will be used to test the efficacy of the derived feature sets, the Bhattacharyya distance, and the χ^2 distance, the former being parametric and assuming Gaussian statistics, the latter



(a)



(b)



(c)

Figure 2.5: Illustration of properties of the DFT which lead to rotation invariance using the filtering scheme. (a) input sinusoids and their spectra below (red pixels). The blue overlay shows the location of Gabor filters superimposed on the sinusoid spectra. The 0° oriented sinusoid's spectra is captured by the first Gabor filter, while the rotated sinusoid spectra falls on the third, and fourth filters. (b-c) Gabor and DFT coefficients of filtered 0° (red), and rotated (green) images.

an attempt to estimate the probability distribution function underlying the data without assuming a specific parametric model.

Bhattacharyya Distance

A well known parametric statistical distance is the Bhattacharyya distance which accounts for the mean and spread of the sample and reference multidimensional distributions [20]. By estimating the mean μ and covariance S, and assuming a Gaussian model as follows

$$D_B = \frac{1}{8}(m_1 - m_2)^T S^{-1}(m_1 - m_2) + \frac{1}{2}\ln(\frac{\det S}{\sqrt{\det S_1 \det S_2}}), \tag{2.18}$$

the distance of each test to each reference distribution (subscripts 1 and 2 respectively) is calculated and a class assignment is made. This parametric model relies on the assumption of normal feature distributions, which is appropriate for Gabor filter features [5]. A major drawback to this model is the need to estimate accurately the covariance, and compute matrix inverses, which for high dimensional data requires large sample sizes. This may not be possible for smaller segmented texture regions, and high dimensional feature sets.

χ^2 Distance

To bypass the unwieldy covariance terms in the Bhattacharyya distance and relax the assumption of Gaussian statistics the probability distribution function of the data can be estimated by forming a histogram. The high dimensionality of the feature sets makes direct construction of the N-D histogram intractable, however the marginal distributions can be estimated. In order to avoid optimizing the bin width and to achieve a more accurate depiction of the underlying marginal probability density functions the smoothing technique of kernel density estimation is used as in [32]. This is a completely non-parametric approach that avoids introducing any (even in the initialization) normality restrictions on the data and has been shown to be more accurate and robust than many previous kernel density estimation methods.

Once the marginal histograms are constructed, the χ^2 distance is used for class assignment as in 2.19

$$\chi^2 = \sum_{i}^{Ndims} \sum_{j}^{Nbins} \frac{(r_{ij} - t_{ij})^2}{t_{ij}}, \tag{2.19}$$

where $Ndims$ is the number of dimensions (and number of marginals), $Nbins$ the number of histogram bins after smoothing, r_{ij} and t_{ij} are the reference and test sample entries at bin j and marginal i.

More Advanced Classifiers

Minimum distance classifiers offer the benefit of low computational complexity, algorithmic simplicity (no parameter estimation or optimization), and can avoid overfitting to noisy data. However, there exist more advanced classification schemes which admit more complex decision surfaces. These allow for class distributions which contain subclasses represented as additional peaks in a class histogram.

The k-NN algorithm uses a majority voting scheme which leads to non-linear decision boundaries and a potential to handle subclasses. It is non-parametric, requiring no knowledge of the dataset's mean or covariance. The major drawback is its lack of power in higher dimensional spaces, since these tend to be sparser and neighborhoods are ill-defined. The naive Bayes classifier is a parametric method which entails estimating the distribution parameters (Gaussian assumptions are typically made) and picking the class for which the test sample has maximum probability. This classifier produces quadratic decision boundaries, and is able to discriminate between samples drawn from ring shaped distributions potentially encompassing other class distributions. This situation would be impossible to accurately classify using a linear classifier. An added benefit of this type of classifier is that it directly calculates the probability the sample belongs to each class. This information can be used as a confidence metric in later evaluation of the results.

2.2.2 Success Validation

To evaluate the success of the classifier there are a number of options, the most obvious being the overall classification accuracy, and the classification accuracy of each individual class. It is also useful to the end user to develop a metric which provides a confidence level for each test sample classified.

Confusion Matrix

Regarding the accuracy of classification, the confusion matrix is a convenient visual aid in investigating accuracies. This square matrix has rows and columns equal to the number

Classes	Class 1	Class 2	Class 3
Class 1	8	2	1
Class 2	1	7	9
Class 3	1	1	0
Accuracies	0.8	0.7	0.9

Table 2.1: Example confusion matrix, with class counts normalized. Each value indicates the percentage that actual Class j was labeled Class i. For example, considering the first column, Class 1 was labeled as such in 8 trials, while in 1 case it was mislabeled as Class 2, and in the remaining case it was mislabeled Class 3. The first row indicates that of the data that were labeled Class 1, 8 were actually Class 1, while 2 were Class 2, and 1 actually belonged to Class 3.

of classes and bins each test classification into the entry corresponding to the row of its actual class, and the column of its given class label. Therefore the quantity in each entry corresponds to the number of samples classified as class i, but labeled as class j, and the count in $i = j$ along the diagonals correspond to the number of correct classifications for that particular class i. This useful visual tool gives quantitative information not only about correct classifications, but also about the distribution of misclassifications, and which classes were most often confused.

Confidence Measures

Once an image has been classified the end user may like to know which of these can be trusted, otherwise a thorough search and manual labeling of misclassified regions is still necessary, and reduces the impact that an automated system will have. The Bhattacharyya distance lends itself to a confidence measure immediately, since a closer distance implies a greater confidence. This distance can be transformed into an error probability known as the Chernoff error using the following,

$$P_i = \sqrt{P(\omega_i)\exp(-d_{Bi})}, \tag{2.20}$$

where P_i is the Chernoff probability of error in the measurement of the distance from a test sample to class i, with prior probability $P(\omega_i)$, and Bhattacharyya distance d_{Bi}. The term error here is misleading, since higher P_i indicates a smaller distance, and a more likely accurate classification.

The Chernoff metric alone does not account for confusion with other class distributions since distance measurements are made pairwise between every test sample and every reference sample. Therefore a comparison of the Chernoff metrics for a particular test sample across all classes may result in a more robust estimate of which test labels can be trusted. To facilitate this, a simple comparison involving the sum of differences between the closest class P_m and the other class distances P_i is made and used as a measure of confidence for each classification. Mathematically this is simply

$$conf_t = \sum_{i}^{Nclasses} (P_i - P_m)^2. \tag{2.21}$$

2.2.3 Data Fusion

The previous sections considered classification using only texture features. The dataset used in this work consists of 3 band RGB images, and the channel intensities may also provide information about the material classes of the surfaces therein. There are many ways to incorporate this information into the classifier, some of which are outlined in the next sections. These three methods will be tested in the following chapters.

Feature Vector Level

The simplest way of incorporating color channel information is to concatenate the three channel intensities onto the texture feature vector. The new feature vector will then consist of the filter responses, as laid out in Section 2.1 and the channel intensities simply tacked onto the end. This method has drawbacks, notably that these separate sources of information must now necessarily be classified with the same distance metric, and some normalization is required.

Filtering Color Channels

Another simple way of classifying color and texture is to perform filtering on each color channel, rather than on a grayscale image. This way any intensity variation in each color channel is captured together. This leads to a very high dimensional space, but avoids the normalization requirements as above.

Voting

An intuitively more plausible approach, voting, allows both texture and color features to be classified independently, and with different classifiers or distance metrics if necessary. The union is then performed on the classified data. In this work we choose three simple methods of voting, given a region that has two independent class assignments.

The first two methods simply take statistics, the mean, and the extremal value from among the two separate classification outputs to determine a "winner". In the case of the minimum distance classifier for each classification, the average distance of the two class assignments is taken, and this is the new distance assignment. Then the region is classified as the class with the minimum distance. The second method takes the minimum of the two separate class assignments, and uses those as the new class distances for each class. Then as before the class with the minimum distance is chosen.

The third method avoids the problem of distances output from various classifier types being unrelated to each other. That is there are no "units" in texture feature space that can be compared to the "units" in color feature space, and therefore comparing the average or extremum distances makes no sense. Rather than working with distances, the Borda count ranks each class according to its distance from the test point, with separate class rankings for all classifiers involved. Next a linear point scale is applied based on the ranking of each class, with highest ranked class (closest distance) awarded the most points (typically equal to the number of classes), and the lowest ranked class (furthest distance) the least number of points (typically 1). The points for each class over all classifiers are then summed, and the class assignment is given to the class with the highest number of points. Table 2.6 shows a schematic example of all three of these voting methods. The data and setup in [26] proves the Borda count to be most successful, but all three methods will be applied in the following chapters.

Color Spaces

Here it is briefly noted there are many color spaces that can be formed from an input RGB image. Because the RGB bands are all highly correlated, much redundant information will be obtained by filtering all of them in that form. The 1976 CIE $L^*a^*b^*$ color space decomposes an RGB response into an orthogonal colorspace consisting of the luminance, a measure of intensity which captures contrast information, and two chrominance channels which capture the color information. The experiments in the remainder of this work will

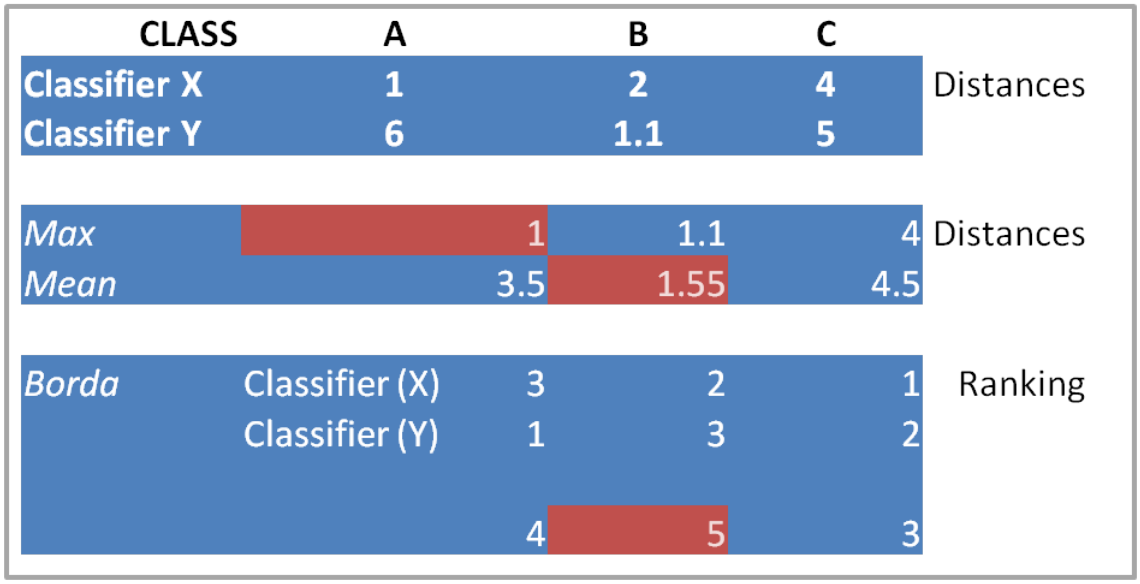


Figure 2.6: Toy example of voting procedure. Two classifiers X and Y determine the distances between a test sample, and three class distributions A, B, and C. Then, three new metrics are calculated for each class - the minimum of the two classifiers measurement, the mean, and the Borda count. The sample is given the label of the class with the new minimum distance, or highest ranking in the latter case. The red highlight indicates the outcome of the vote for each method.

utilize the $L^*a^*b^*$ space because of its separability.

Chapter 3

Experimental Design and Setup

This chapter will discuss the details of the datasets and experimental procedure used to investigate the proposed classifier. As laid out in the previous chapter, the Gabor filtering scheme leaves room for some free parameters, and there are several methods to combining color and texture information. These will first be investigated on commonly used databases of texture images composed of both manmade and naturally occurring materials that mimic those found in real world scenes. The results of the classification will inform the final configuration of the classifier, and the real world dataset of the Purdue campus will be described as well as the experimental procedure for this dataset.

First, both texture databases will be described in detail, and later the real world imagery will also be detailed. Following this, the experimental procedure to test the classifier on both the assembled databases and real world imagery is laid out. There are several practical questions regarding implementation of a classifier like this that are answered. These include how to best incorporate color information given the varying illumination conditions, how much training data is necessary to cope with the variation in sample appearance (due to illumination changes, and within class variation), and at what step to amalgamate the data such that a homogeneous region is labeled, rather than individual pixels. The first and second questions can be answered by testing the classifier on the two texture databases, and the last question on the Purdue dataset.

3.1 Texture Databases

Two texture databases were used in the experiments described in the following sections to test the configuration of the filter bank, the methods of incorporating color information, and the two minimum distance classification algorithms. First, the Brodatz database [33] was used to test the performance of the rotation invariance of the feature set, and its sensitivity to bandwidth. The Curet database [34] was then used to investigate the addition of color channel information, and test the classifiers' sensitivity to variations in look and illumination angle.

3.1.1 Brodatz

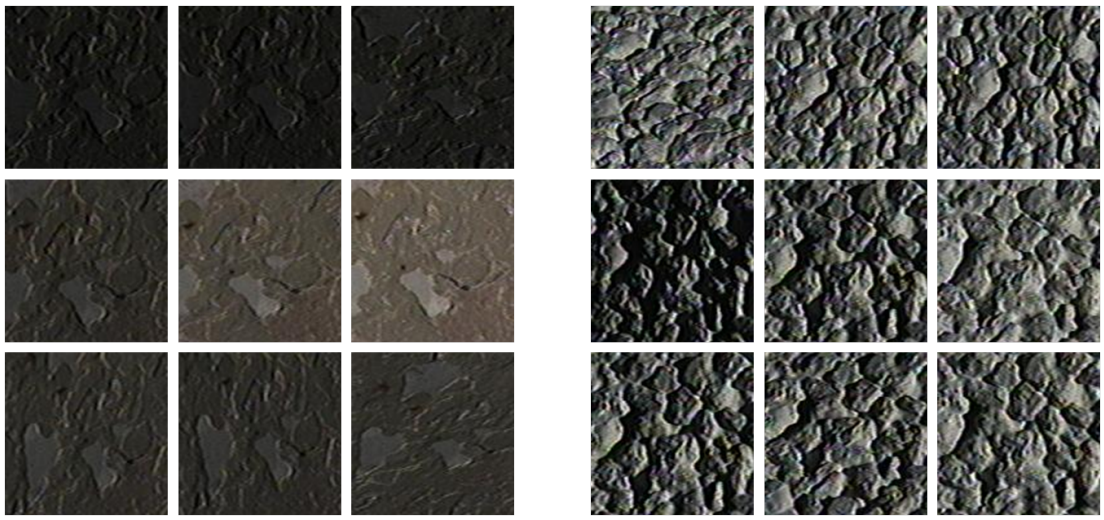
The Brodatz database is a photographic album consisting of grayscale images from the original work of Brodatz [33] and is commonly used in texture analysis experiments [16, 6]. The album is made up of over one hundred texture samples photographed directly overhead under controlled studio lighting conditions. Some of the photographs are zooms of identical textures, but this database is one of the most diverse collections of textures available. There are oriented, stochastic, and periodic textures that span a wide variety of scales, and shapes. Figure 3.1 shows the subset of this database that will be used in this work. This subset of 30 images was taken to eliminate any textures repeated at different zoom levels, and for computational ease. The 8 bit digitized images used in this work were taken from [35]. This database will be used to test the rotation invariance of the feature set, without incorporating any color information.

3.1.2 Curet

The second texture database that will be used for the majority of the testing before coming to the real world dataset is the Curet database [34], which is publicly available online for browsing and download. This remarkable database is a set of over two hundred BRDF measurements of over sixty different real materials. The materials include both naturally occurring and man-made objects containing stochastic, oriented, line like, and blob like textures. Each material was illuminated (with a constant source) and captured from a variety of angles to build up a BRDF function. The RGB images at each illumination and view angle are made available as 24-bit files.

This database is useful for the work herein because of the range of view and illumination

Figure 3.1: Sample images from the Brodatz album used in this work.



(a) (b)

Figure 3.2: Sample images from the Curet album used in this work. (a) Limestone class, (b) Concrete class.

angles which simulate the changing illumination conditions of the aerial imagery. In this work, a subset of the database consisting of 17 interesting materials that would be found in urban and residential buildings was taken, and is shown in figure 3.2. Unfortunately, this database does not entirely simulate the conditions found in aerial imagery, due to the much higher resolution. Many of the textures are quite rough, and especially at shallow incidence angles exhibit self shadowing and other 3D effects that would not be apparent in aerial imagery. To mitigate this to some degree the subset found in [36] is used, which excludes angles less than 60°. The images were also cropped to 200 by 200 pixels, for processing speed. This database will be used to test various configurations of combining texture features and color channel information.

3.2 Real World Dataset (Purdue Campus)

3.2.1 Capture Conditions

The real world imagery available to test the classifier is a set of RGB images captured over the Purdue University campus in Purdue, Indiana. These images were taken by Pictometry Inc. on an aircraft equipped with four cameras oriented approximately orthogonally to

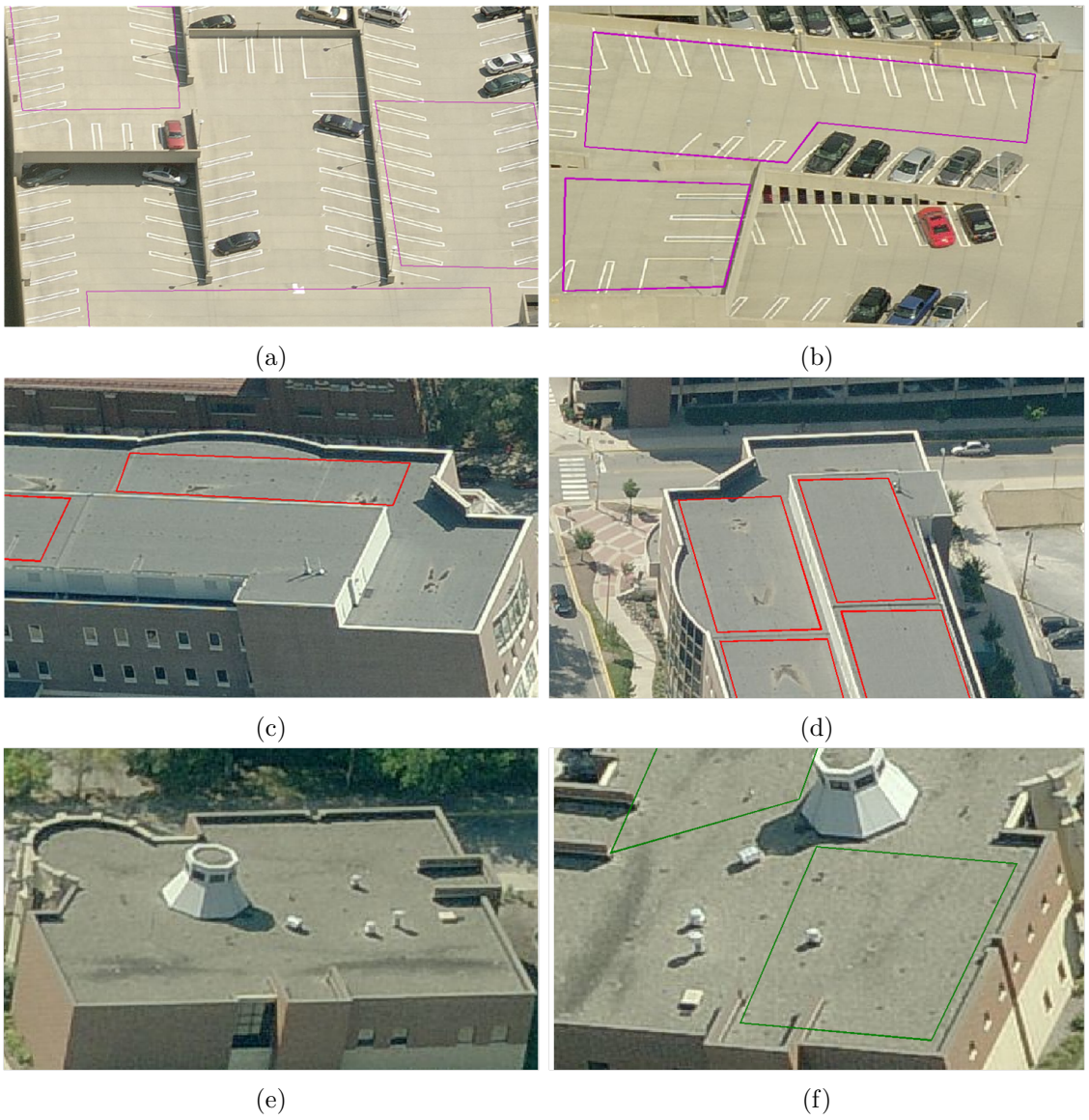
each other, and pointed toward each of the cardinal directions. These are set at an oblique angle of approximately 45° A fifth camera is used to capture a nadir view. Thus a complete 360° view of every scene is generated, and building façades are readily available. This dataset could nominally allow full classification of building façades and rooftops. The challenges of this dataset as illustrated in Chapter 1 include varying GSD away from the immediate look angle, affine distortion of texture in the image, and varying illumination at different oblique look angles.

The Purdue campus was captured at three separate craft altitudes, resulting in three datasets with decreasing GSD of 6, 4, and 2 inches. At the highest GSD only the aggregated effects of textures such as roof ballast are visible, while at 2 inches individual roofing shingles and clay tiles are nearly discernible.

3.2.2 Scene Content

Figure 3.3 shows sections from two cardinal directions for each resolution. The differences in appearance due to the changing view angle, and the various material BRDF's are immediately discernible. The Purdue campus consists of residential buildings, larger academic buildings, and parking garages. For the purposes of this work, to test the efficacy of a standard classifier on oblique imagery, we will focus on the building rooftops. The roof classes were used because they contained a diversity of man made materials exhibiting a variety of color and texture content. They are also more homogeneous than building façades and more easily segmentable due to less occlusions.

Six classes of roof were identified that appeared frequently in all images; these are residential shingles, red clay tile, aggregate stone (ballasted roof system), bare black membrane, bare White Membrane (TPO), and parking garage. Figure 3.5 shows examples of the 6 classes at each GSD. These classes represent spectral diversity and contain varying levels and patterns of texture. The clay tile is distinctly red and contains a linear pattern that is readily visible at 4 and 2 inch GSD. The residential shingled rooftops vary in color, but are typically darker colored and exhibit a semi-regular rectangular array that is more stochastic than the clay tiles. The rubber membrane class is dark in color, and has a fine grained matte appearance, but at the lowest GSD some of the seams between each layer become visible. The same is true for the White Membrane (TPO) roofing system, and in addition both of these vary substantially in the level of dirt and erosion present. The parking garage class primarily exhibits texture due to the yellow lines which are reg-



(a) (b)

(c) (d)

(e) (f)

Figure 3.3: Example regions from images of varying look angle. Classes shown are "Pk" (a-b), "Rubber" (c-d), and "Ballast" (e-f).

ularly spaced. The orientation of these lines varies even in the same image. Finally, the ballasted roofing system consists of loose stone of varying grain size scattered over the rooftop. These roofs exhibit stochastic texture rougher than either the white or black membrane roofs.

Figure 3.6 shows the variation in frequency spectra for two classes as the resolution and image orientation is changed. Since the classifier is designed to respond to variations in the spectra these are significant. The dominant building orientation produces a strong peak in many of the spectra. Although the boundaries of the homogeneous texture regions will be masked there will still be edge effects present in the feature images, which will be enhanced for filter based methods which require a large region of support. This is one of the major drawbacks of using this method. It is also apparent that the image scales change, even within the same image from the center to periphery. The dominant textural components shift in both center frequency, and orientation.

Another issue is the class representation. For the 6 and 4 inch GSD several samples are available of each class in one image, while for the 2 inch resolution there are fewer class examples for parking garage, and the White Membrane (TPO) class.

3.3 Texture Database Setup and Experiments

Both the Brodatz and Curet databases described above were used to tune the classifier's free parameter, the bandwidth, as well as test the two minimum classification methods - the Bhattacharyya and χ^2 distances. The Brodatz database was used to test the classifier's invariance to rotation, and the Curet database was used to test the several ways of incorporating color content into the feature set.

3.3.1 Database Setup

Brodatz

The 30 classes from the Brodatz album chosen for these experiments are shown in Figure 3.1. These were first cropped to 512 by 512 pixel size, and for each class, six additional tiles were produced by rotating the initial orientation by 15°. This produced a total of seven tiles per class, identical except for orientation, which with six rotations resulted in orientations from 0 to 90°. Since some classes were horizontally symmetric, 90° was the

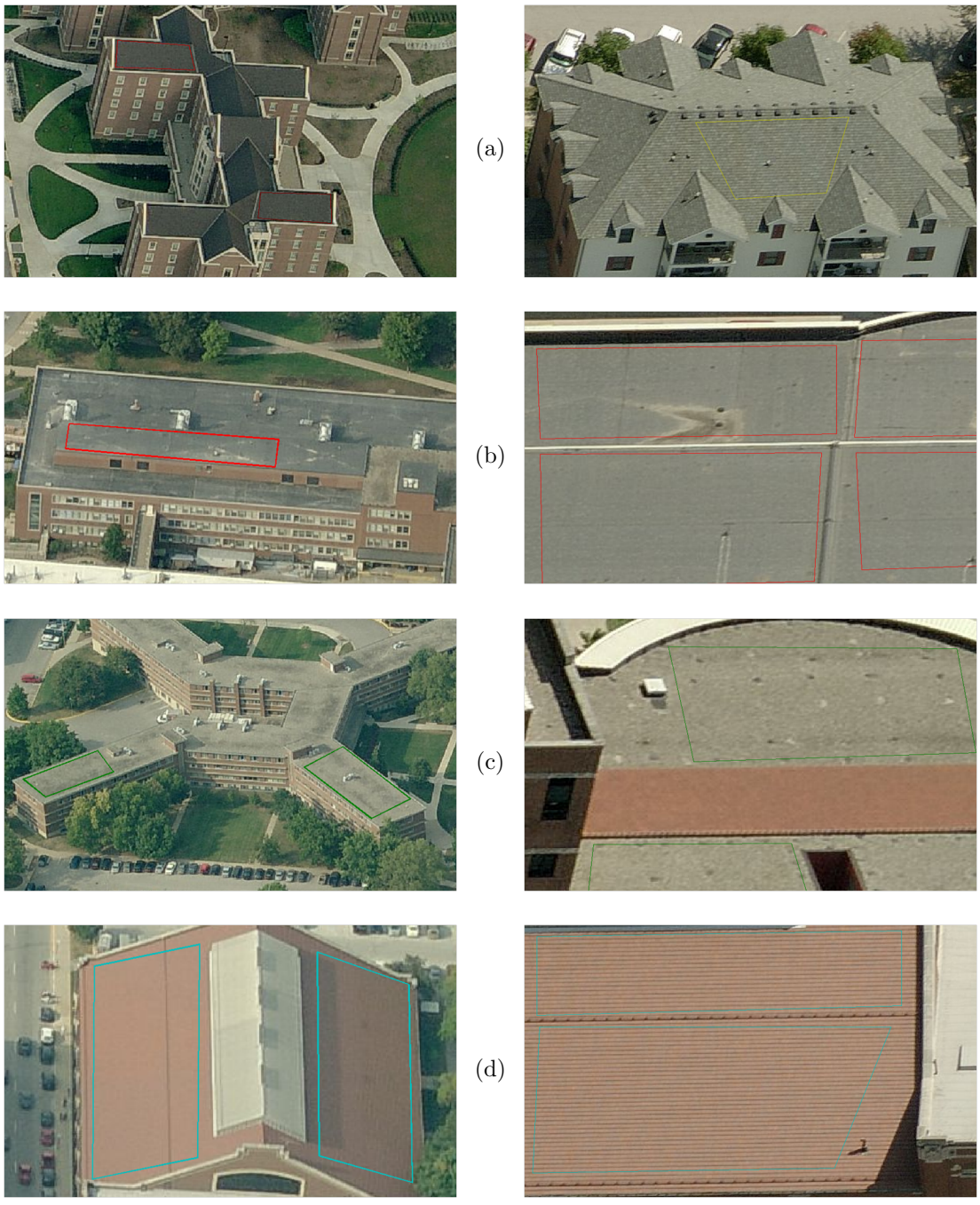


Figure 3.4: Example regions from the Purdue dataset at 6 in GSD (left column) and 2 in GSD (right column). Classes shown are, (a-d) “Shingle”, “Rubber”, “Ballast”, and “Clay Tile”.

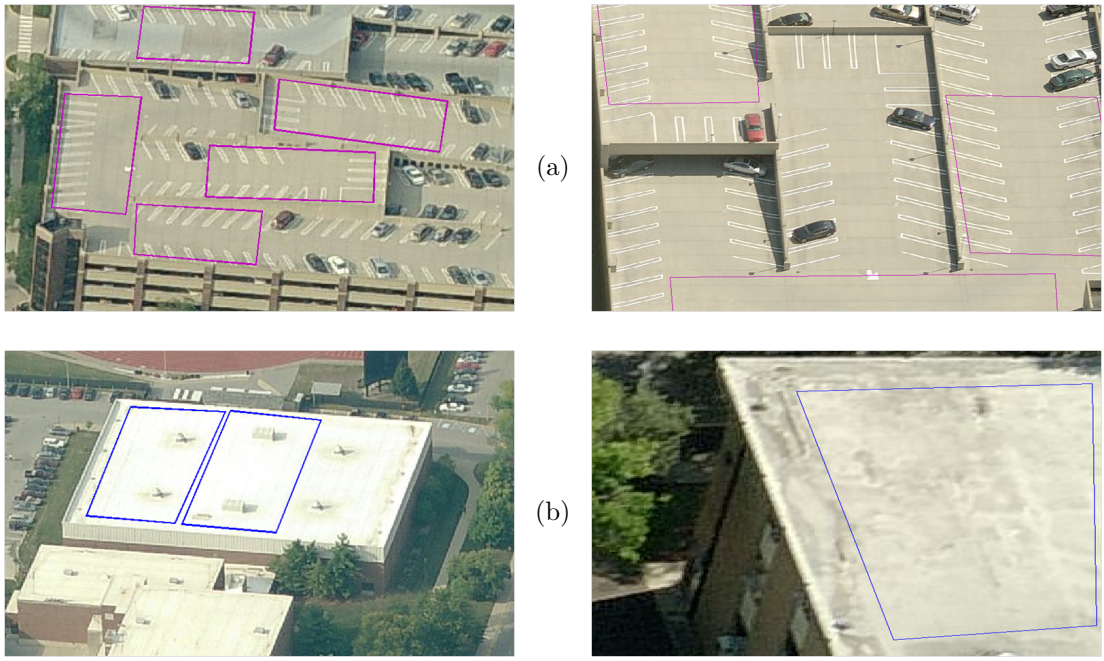


Figure 3.5: Example regions from the Purdue dataset at 6 in GSD (left column) and 2 in GSD (right column). Classes shown are, (a-b) respectively, “Parking Garage” and “White Membrane (TPO)”.

cutoff, rather than 180°. After rotation, the 512 by 512 images were cropped to 128 by 128 pixels for computational ease. Figure 3.7 shows examples of the rotated tiles.

Curet

As stated above 17 classes were chosen from the Curet database with 92 images per class with illumination and view angle varying from nadir, to 60° off-nadir. Each image was cropped to 128 by 128, and these images were then divided into 32 by 32 pixel image tiles, producing a set of 92×16 image tiles per class available to be used in classification. This was done so that each class could be sampled uniformly over the range of variation both of the texture itself, and the illumination and viewing conditions. Due to processing limitations, a subset of 20 of the 92 images differing in view and illumination angle were randomly chosen for each classification.

3.3.2 Classifier Details

Texture Features

As described in Chapter 2 the classifier consists of a bank of filters centered and oriented to provide maximal coverage of the frequency domain. The center frequencies used were $\sqrt{2} \times [1, 2, 4, 8]$ cycles/image, with a filter window size of 32 by 32 pixels. At each center frequency six circularly symmetric ($\sigma = 1$) filters were used corresponding to orientations of 0 to 150°. The frequency bandwidth was allowed to vary from 0.7 to 1.5 octaves, and is specified in later experiments. For a given set of filter responses the absolute value was used as a feature along with its spatial variance. These were then made rotation invariant as outlined in Chapter 2 by taking the first four features of the DFT along the orientation axis of both absolute value, and its variance. For each image, the filters were applied in Matlab in the spatial domain via convolution assuming periodic boundary conditions. The images were not preprocessed.

Color Features

For the Curet database the color components were added using three different methods as detailed in Chapter 2. For comparison, the texture features were also used alone by transforming the images to grayscale. For the color methods, the image was first transformed to $L^*a^*b^*$ space. The first color image method was to concatenate the texture

feature vector with the color channel information. First the texture features were extracted at every pixel, then the color features were added. The next method was to filter every color channel independently, and concatenate these into a single feature vector to be used for classification. Finally, the third method was to classify the texture feature images, and color images separately, then combine them using the voting methods outlined in Chapter 2.

Classification

The minimum distance classifiers outlined in Chapter 2 were both used for the Brodatz, while only the Bhattacharyya distance was used for the majority of the Curet database trials. To use the Bhattacharyya distance, training data from the feature space of each class was selected accordingly, and added to each class class distribution. For the χ^2 distance the feature spaces were then transformed into a 256 bin normalized histogram using the kernel density estimation technique described in Chapter 2.

3.3.3 Texture Database Experiments

Brodatz

The grayscale Brodatz database was used only to test the rotation invariance of the classifier texture features. The color channel intensities were assumed not to be rotationally invariant. Using this database, training data for each class was selected from among the 0° oriented images, and the task of the classifier was to classify the remaining six orientations into the proper class. The bandwidth was left as a free parameter, and after the initial round, an optimized value was chosen to look at individual class accuracies, and orientation angle accuracies.

Curet

Initial trials with the Curet database were aimed at determining the amount of training data necessary to account for all of the variation at one view and illumination angle, and over a range of illumination and viewing conditions. For each classification, an increasing percentage of the 20×16 tiles were chosen as training data, and the rest used for validation. First, only the texture features were used and all images were transformed to grayscale by weighted addition of the color components. Then, in turn each of the three methods

for incorporating the color content were used. Because the method of classification by comparing histograms was shown in the first few experiments to be poor, it was only used with the grayscale, and one color method. For most of these experiments the bandwidth was kept as a free parameter, and was shown to have a different optimal value depending on the method of incorporating color content.

3.4 Purdue Setup and Experiments

3.4.1 Purdue Setup

Data

Use of the Purdue imagery as described in Section 3.2 above proceeded first by manual segmentation of the images into the six roof classes identified. Care was taken to select homogeneous regions, though this was not always possible due to rooftop structures, and erosion and weathering of the materials. With the 3D data, building rooftops and façades could in principle be automatically segmented. Additionally, further image segmentation based on contrast variations could delete clutter and other unimportant rooftop structures to create a more homogeneous region, but this was not done here.

All three of the image resolutions available, with GSD as described above were tested. For each resolution several images from two of the sensors were chosen such that training and validation data could be taken from across different look angles. Table 3.1 summarizes the images that were chosen at each resolution. The number of images and look angles available were limited by the frequency of the chosen rooftop classes. Table 3.2 shows the occurrence of available classes for each resolution, cardinal direction, and image number. Because of the high resolution there are less ROIs available for the parking garage and White Membrane (TPO) classes at the 2 inch GSD. Figure 3.8 shows a sample image of all the regions from each class taken from a North facing image from the 6 inch resolution.

Classifier

The same setup for the classifier is used here, except that only the preferred method of incorporating color information was used, and the feature images were classified using the Bhattacharyya distance method as it was discovered early on that the histogram method was not suitable, and also slower. The results in the following chapter for the Curet

Table 3.1: Number of images available for each resolution and look angle, expressed as abbreviated cardinal direction.

GSD [inches]	6	4	2
Look Angle	# Images per Look Angle		
N	2	0	4
S	0	2	0
E	2	2	4
W	0	0	0
# Images total	4	4	8

Table 3.2: Number of images available for each resolution and look angle, expressed as abbreviated cardinal direction.

GSD [Inches]	Cardinal Direction	Shingle	Clay Tile	Ballasted	Black Membrane	White Membrane	Parking Garage
2	E	11	8	3	7	2	5
		11	8	4	5	3	5
		9	11	6	6	2	1
		0	9	8	1	0	0
	N	4	18	7	5	2	2
		11	14	9	7	9	0
		6	12	7	2	5	0
		1	13	6	3	2	0
4	E	9	13	11	6	3	7
		7	16	10	5	9	2
	S	9	13	12	6	4	6
		6	13	9	5	9	0
6	E	12	11	13	11	10	5
		12	11	18	10	10	5
	N	11	12	11	9	9	5
		13	13	14	11	7	8

database experiments show that the classification is not extremely sensitive to the bandwidth, nevertheless its value will be optimized for each resolution. In a semi-automated approach this could easily be done prior to classification using a subset of the training data for validation. For the experiment comparing data clustering the Mahalanobis distance is used for distance calculations involving a single feature vector and a class distribution.

3.4.2 Purdue Experiments

The experiments to be performed on the Purdue dataset are generally designed to determine how standard texture features fare on highly variable class distributions due to the changing illumination and view angles of oblique imagery. It is also desirable to know whether regions in an image from one look angle can be used to classify regions from another image that may be from a different look angle and how much training data is necessary for success. Finally, it is useful to understand the implications of clustering the data, and whether to first pool single feature vectors into a distribution and classify, or classify individual pixels and then segment them into homogeneous regions.

First experiments

The first experiments used from one to seven training regions from one image at a particular cardinal direction and resolution and classified all of the regions present in another single image. Training regions were randomly selected from the regions available for each class, and each classification was repeated several times to look at the impact of varying training regions. The overall classification accuracy was primarily used in the assessment of performance. Using this metric the performance of training and classification in the same cardinal direction, and across different cardinal directions and resolutions was assessed. In addition the variance in individual class accuracies as different training regions were used was noted.

Confidence Metric

A true confidence metric was unobtainable with this particular minimum distance classifer, however the relative distances of a sample region to each class gives some indication of the confidence of that classification. A simple metric was calculated for each sample classified by converting the Bhattacharyya distance into a Chernoff error, equation 2.20, which

accounts for the class priors. Then for each sample, the ratio of the minimum distance to each of the six other class distances was taken, and these values summed. This gives a pseudo-confidence metric that is dependent to some degree on the samples' proximity to each class center, and the inter class distances as well. This metric was then used to threshold the classified test samples and throw out successively more samples classified with a low confidence level, until ideally only samples that were classified with a high confidence level (presumably correctly classified) remain. The efficacy of the confidence metric can be observed by noting how many of the classified test samples need to be thrown out until the remaining pool is classified with 100% accuracy.

Aggregating Classified Data

Once a feature image has been generated by the filtering and DFT normalization procedure, there are two methods of proceeding such that an entire homogeneous region has been labeled, rather than just individual pixels. First, since the feature image should already correspond to a somewhat homogeneous region from the prior segmentation informed by the 3D registration, this feature image as a whole could be classified using the Bhattacharyya distance. Thus the entire region is treated as a distribution, and its proximity to each class distribution is compared. The second method is to classify this feature image using the Mahalanobis distance pixel by pixel, thus the regions' individual pixels are labeled. Following this, these class labels can be aggregated in some fashion to form a homogeneous region label. Both of these options were tested, using the Bhattacharyya distance for the first case to immediately classify the entire region, and using the Mahalanobis distance in the second, then simply giving the entire region the label of the class which occurs most often within the region. This is admittedly an unintelligent method, but it is used simply for comparison purposes to get insight into the logistical question raised regarding the order of aggregation.

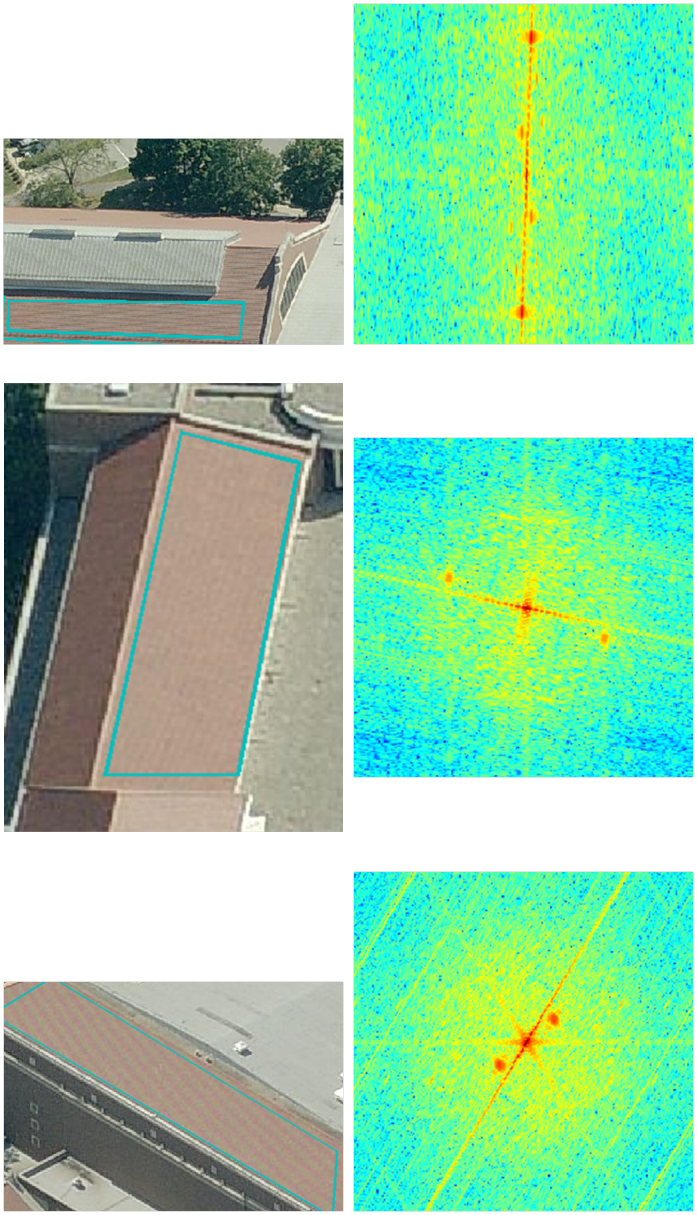


Figure 3.6: Clay tile roofs from several viewing angles (left column) and FFT (right column). The outline shows the masked region. The masking produces edge effects which are manifested in the FFT as continuous lines. The dominant spectral features from the texture itself can be seen as localized symmetrical spikes around the origin. Their center frequencies and orientations change as the scale and orientation of the region changes.

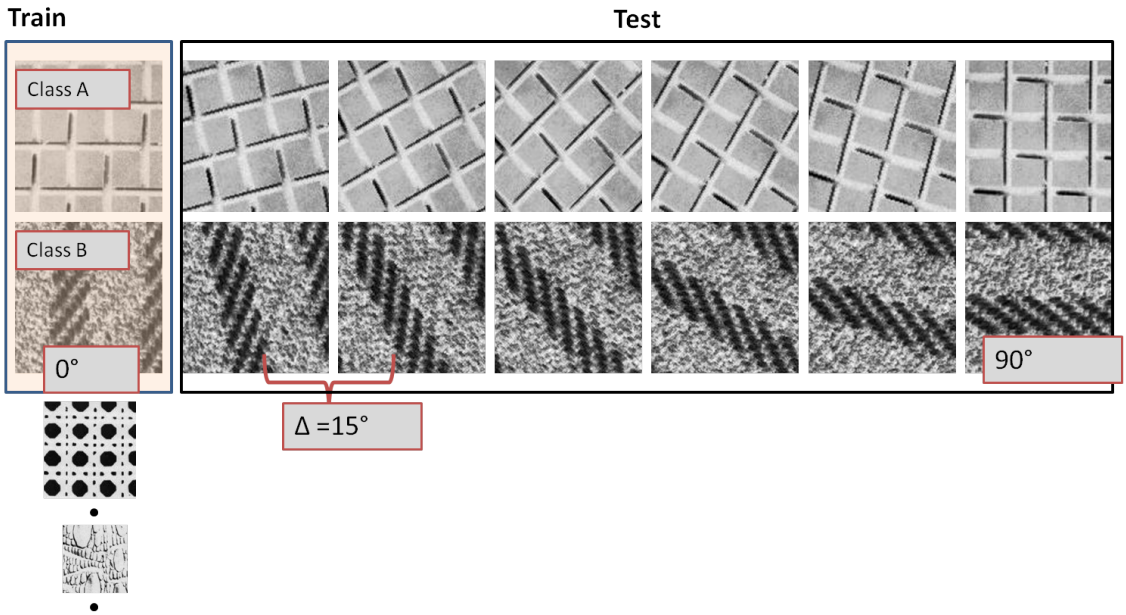


Figure 3.7: Setup for test of rotation invariance. Each original Brodatz image was rotated, then cropped to a 128×128 tile. The classifier was trained on the initial 0° orientation and used the 6 rotated tiles as test set.

Figure 3.8: Purdue 6 inch rois, N facing image.

Chapter 4

Results

4.1 Brodatz

The initial tests run on the Brodatz database outlined in Chapter 3 were designed to test the rotational invariance of the classifier. This section underscores the major results of these tests.

4.1.1 Overall Accuracies

First the overall accuracy was examined. This is the total number of correct classifications divided by the total number of attempted classifications. This metric gives an average value for classifier performance over all classes. The overall sensitivity of the classifiers to bandwidth and at each rotation angle are plotted in Figures 4.1 and 4.2 respectively.

Bandwidth

Figure 4.1 shows that for increasing bandwidth the classification accuracy shows an upward trend, with a dip in the Bhattacharyya classifier performance at $B = 1$. The χ^2 distance appears to be less sensitive to the choice of bandwidth. The bandwidth should ideally be optimized before a new dataset is classified because it is unknown which and how many frequencies within a spectral region will allow sufficient separation in feature space. For these experiments the success of higher frequency bandwidths (lower spectral, higher spatial resolution) may indicate that higher spatial resolutions are preferred at the expense of spectral resolution.

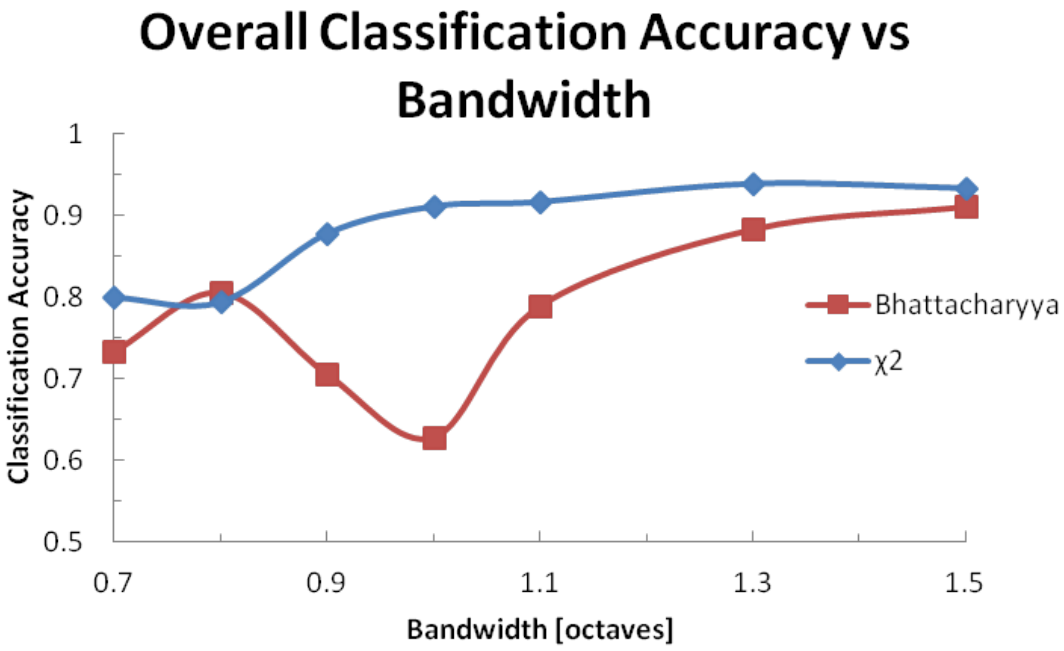


Figure 4.1: Effect of filter bandwidth on classification accuracy of rotated Brodatz tiles.

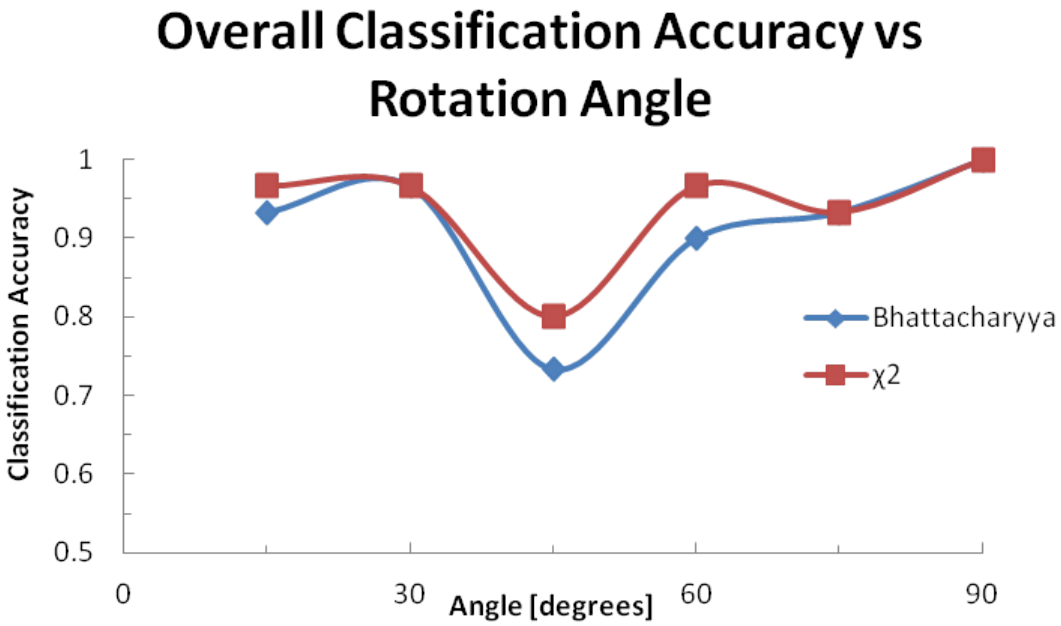


Figure 4.2: Overall classification accuracy of rotated Brodatz tiles at each rotation angle.

Rotation Angle

Figure 4.2 shows the classifier's sensitivity to each discrete rotation angle. The dip in accuracy is predictably at 45° where the images most distinct from the 0° orientation are produced. Both of the classification methods perform equivalently well, remaining at over 90% accuracy except for the dip.

4.1.2 Class Accuracies

Figure 4.3 shows the performance of each class of texture tested. The accuracy is indicated by the colored outlines. About half of the textures were perfectly classified, and all but two were correctly classified in all but one case. The worst cases were classified correctly classified at half of the rotation angles.

The above investigations of the classifier's performance when only rotation is a factor indicate that it can sufficiently handle rotations of oriented and stochastic textures to arbitrary angles, and almost flawlessly to angles less than 45°. The classifier still has difficulty discriminating rotation of textures which contain purely one dominant orientation.

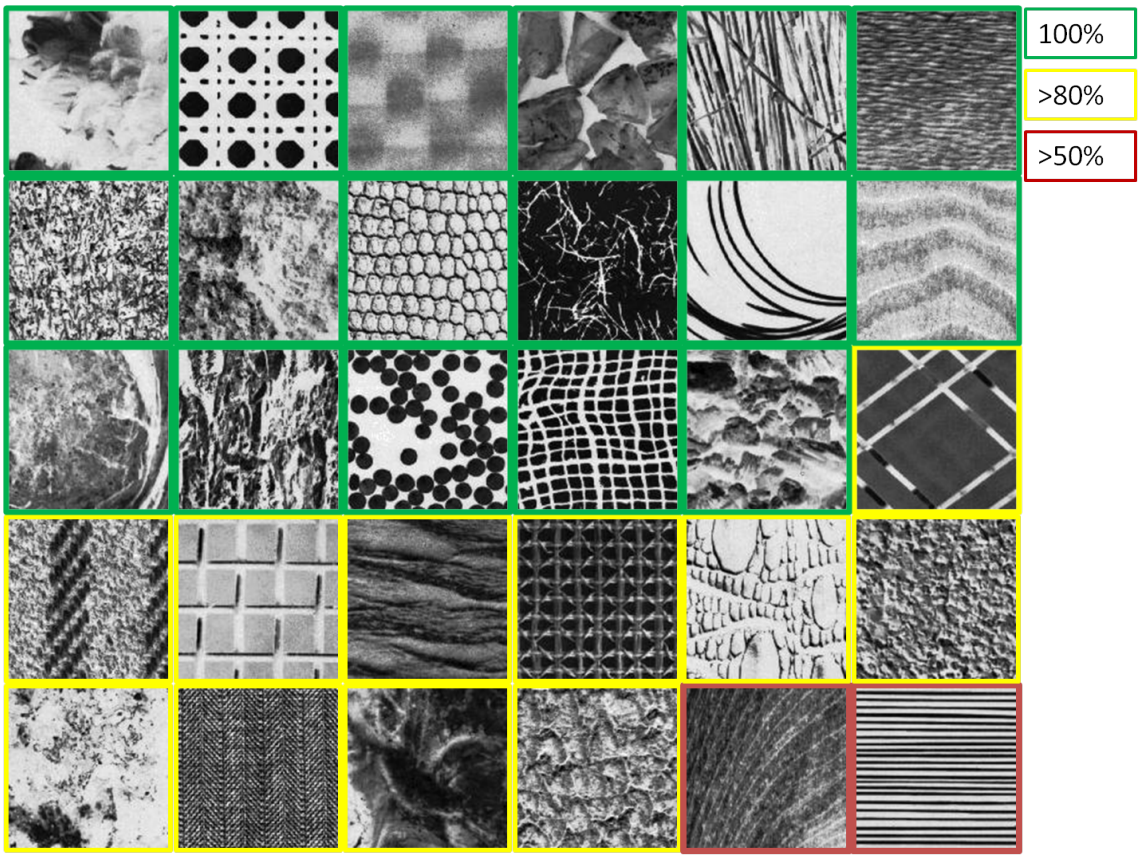


Figure 4.3: Average classification accuracies for each Brodatz class.

4.2 Curet

This section details the results of classification on the Curet database, outlined in Chapter 3. Both the overall accuracies and the individual class accuracies are observed. The purpose of these experiments was to determine the best method of incorporating color content along with the filter responses, and the percentage of training data necessary to achieve sufficient accuracy.

4.2.1 Overall Accuracies

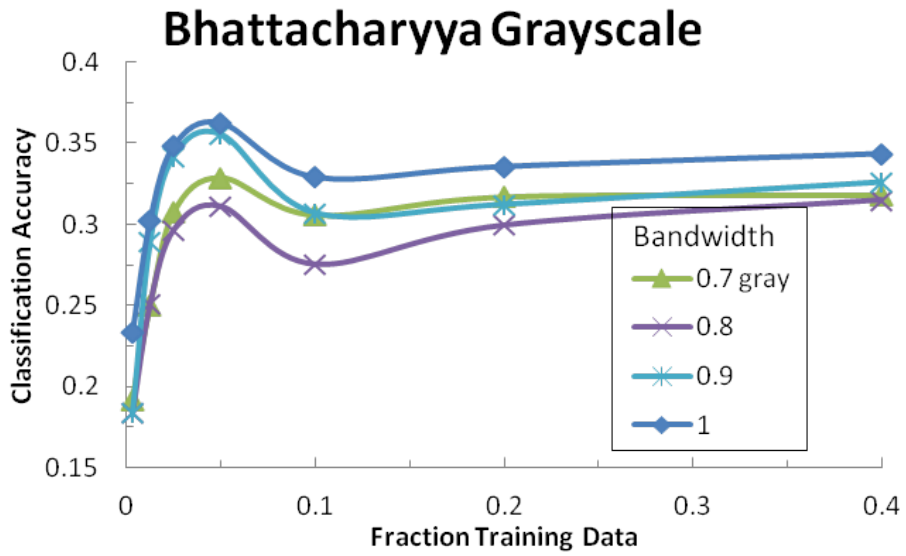
Bhattacharyya Distance

Figure 4.4 and 4.5 show the overall accuracy achieved as a function of the percentage of training data using the Bhattacharyya distance classifier along with three different methods of adding color information. Also shown is the result of using the Bhattacharyya based classifier on grayscale images. For each method the bandwidth was also varied. The results seem to be largely insensitive to the choice of bandwidth, thus the next section's results will be shown at a single bandwidth.

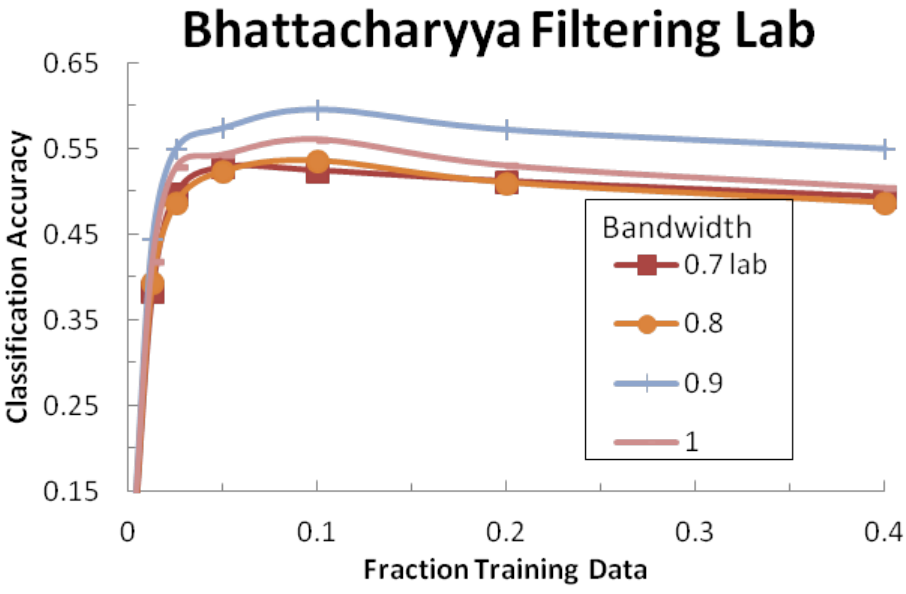
The classifier performed poorly without the addition of color content, achieving a maximum accuracy of near 35%. Adding the color information boosted performance to near 60% no matter which method was used. Despite the much larger feature size of the first method (filtering every color channel) it barely outperformed the much simpler and inexpensive method of concatenating the color channel intensities onto the filter response feature vector.

There was little difference between utilizing all three $L^*a^*b^*$ channels, and using only the chromaticity components on this dataset. Figure 4.4 shows that for a training subset of less than 5% the results are slightly better for the case of appending only the chromaticity components than including the luminance component as well. In the case of training and testing data which exhibits changes in illumination it is expected that the luminance component will add erroneous information since it varies with overall contrast level. The chromaticity components are much less sensitive to these variations and thus may be better suited to changes in illumination levels.

The performance over a range of training data indicates that for this classifier, a minimum of about 5-10% of (disjoint) training data should be used. Beyond that, overfitting appears to decrease results, and is bound to happen as the pool of test data shrinks.

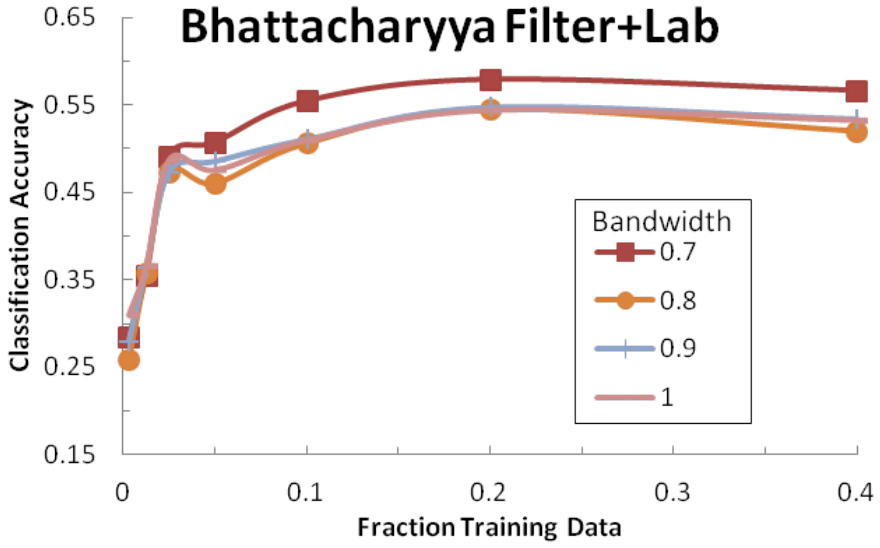


(a)

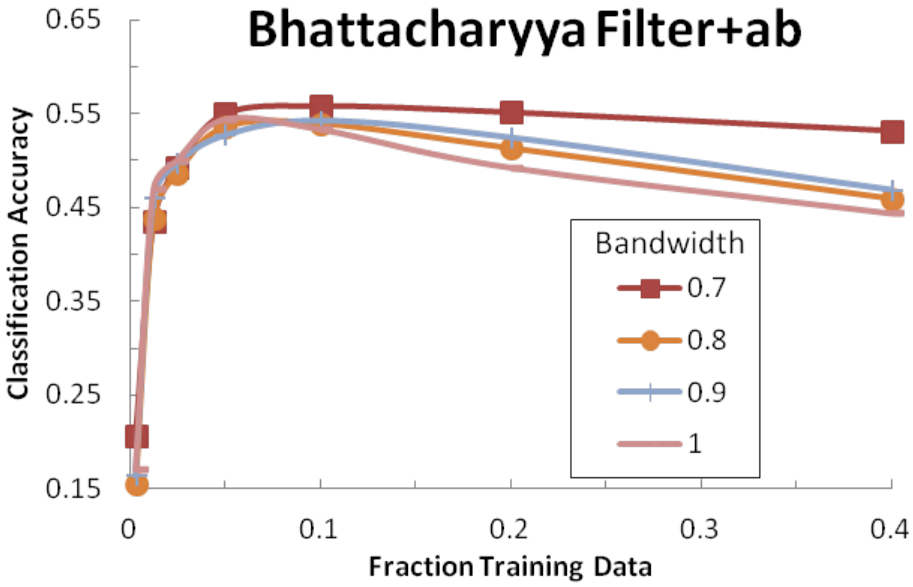


(b)

Figure 4.4: Classification with Bhattacharyya distance. (a)-(b) respectively show filtering grayscale images and appending $L^*a^*b^*$, and a^*b^* intensities onto the overall feature vector.



(a)



(b)

Figure 4.5: Classification with Bhattacharyya distance. (a)-(b) respectively show results using feature sets derived from grayscale images only, features generated filtering all color channels. Each case was performed with the four bandwidths shown.

χ^2 Distance

Figures 4.6 and 4.7 show the results of utilizing the same feature sets with the χ^2 minimum distance classifier. Only the first few runs were performed under the assumption that the accuracy would peak and fall off. Given the very low accuracies at small portions of training data and the apparent early falloff it was clear that regardless of the method of incorporating color content, packaging the filter and color responses as a concatenated marginal histogram led to much poorer results than using the Bhattacharyya distance.

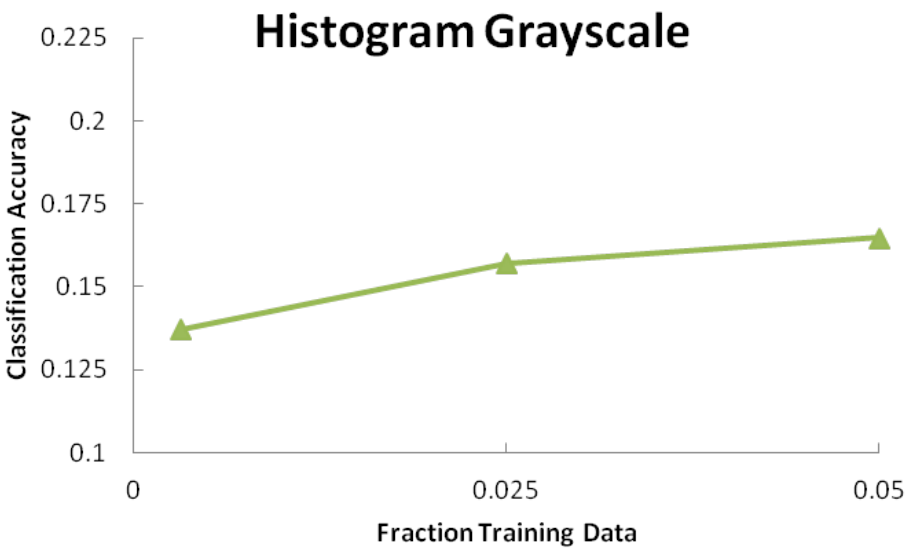
Voting

Figure 4.8 shows the overall classification accuracies by combining the filter responses, and color channel intensities using three voting methods. By combining the classifier outputs in this way accuracies similar to the other methods of utilizing color information are achieved, upwards of 55%. This methods appears to be insensitive to the amount of training data; beyond about 5% inclusion it reaches a steady state. The best voting method is the Borda count, most likely because it is non-parametric, and normalizes out incompatible scales through the ranking procedure. This is in agreement with [26]. Figure 4.8b shows the difference in classification accuracy (in percentage points) between a single classifier system and the dual classifier voting scheme. The Borda count method shows an improvement of about 10 percentage points.

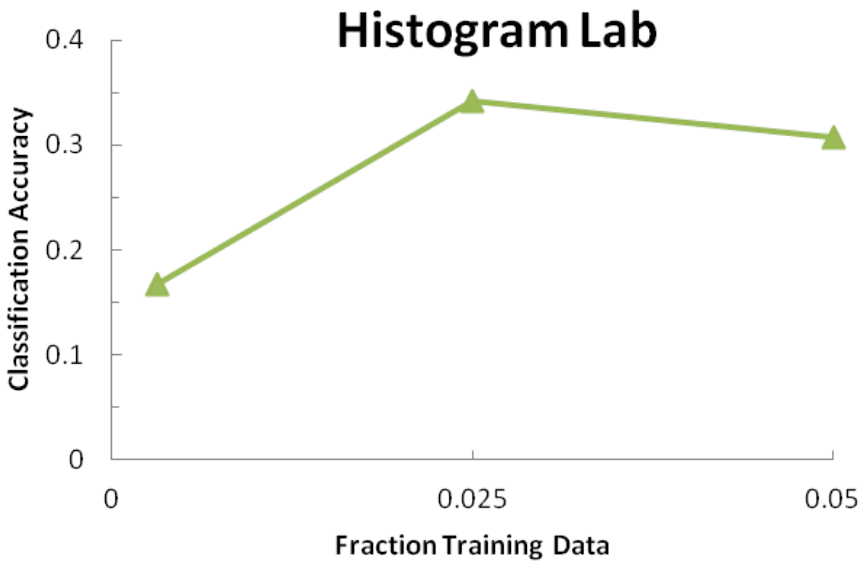
The individual efficacies of the color components and filter responses are shown in Figure 4.9. It is shown that the color components are modestly superior when used alone in comparison to the filter responses. However, the grayscale filter responses show less sensitivity to incorporation of new training data and less tendency to overfit. This is probably due to the erroneous information that is embedded into the color channels as the BRDF angles change.

4.2.2 Class Accuracies

Figure 4.10 shows classification accuracies typical for each class using the above classification methods. It appears that the color channel information was used to a high degree given that the top classes were distinct in spectral character, and the worst seven classes were very similar in that regard. The top four classes were the most finely grained and stochastic. However, it is difficult to draw conclusions from this dataset regarding the



(a)



(b)

Figure 4.6: Classification on grayscale images (a) and color images (b) using the histogram method.

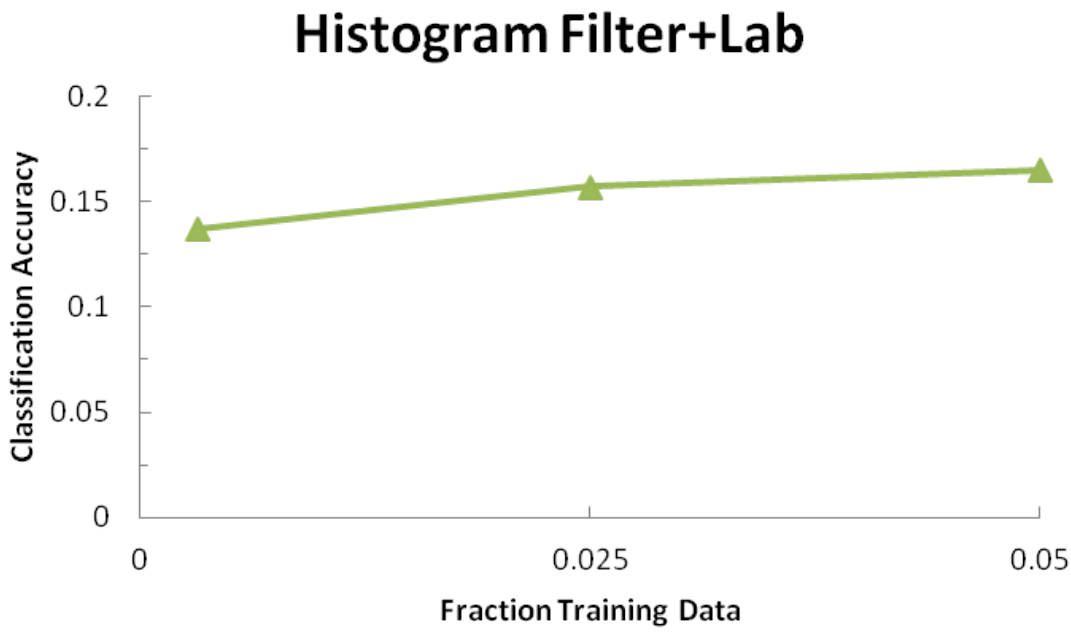
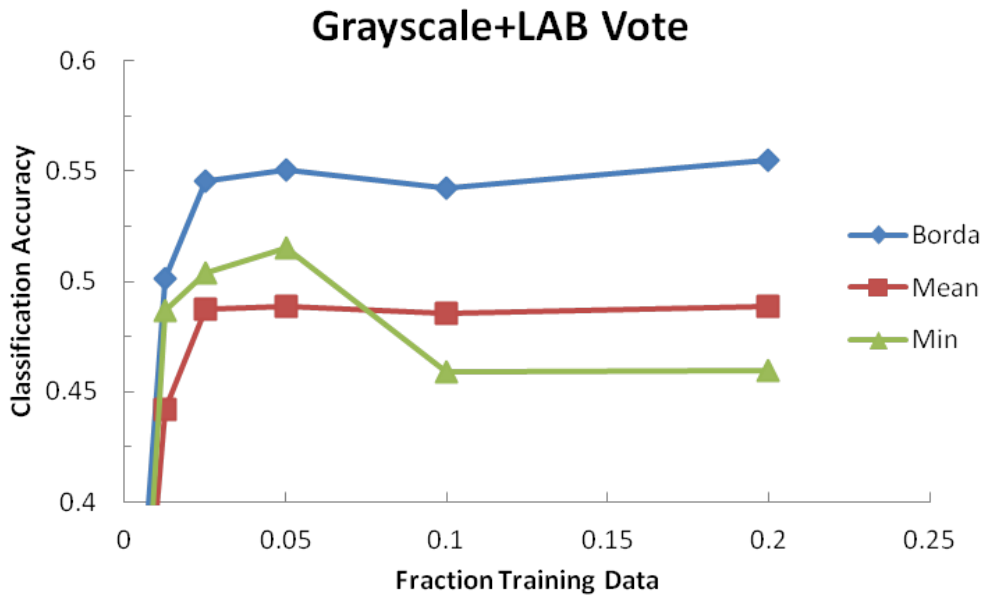
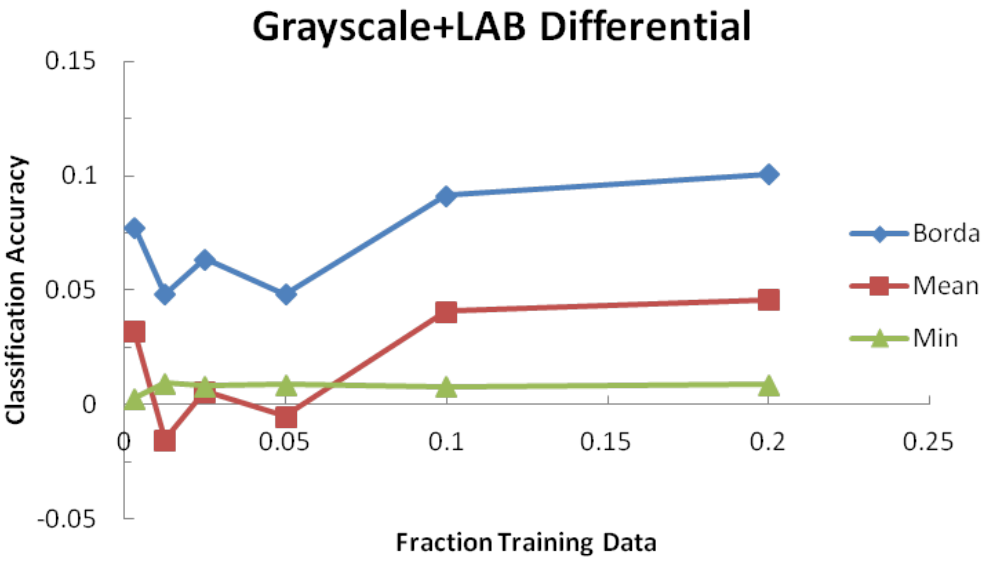


Figure 4.7: Classification of features derived from filtering grayscale images and appending color channel intensities to the feature vector.



(a)



(b)

Figure 4.8: Classification results using the voting methods. (a) shows the overall accuracy, (b) shows the gain (or loss) in classification accuracy after combining the classifier outputs.

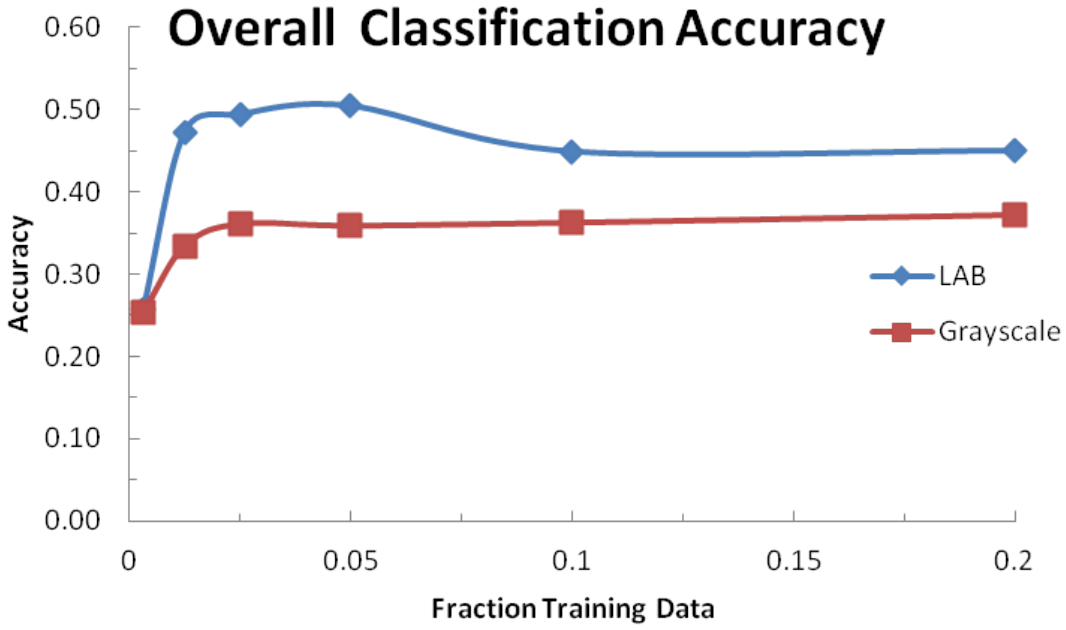


Figure 4.9: Overall classification accuracy on Curet images for two classifiers, one using only color channel intensities, and the other grayscale filtering.

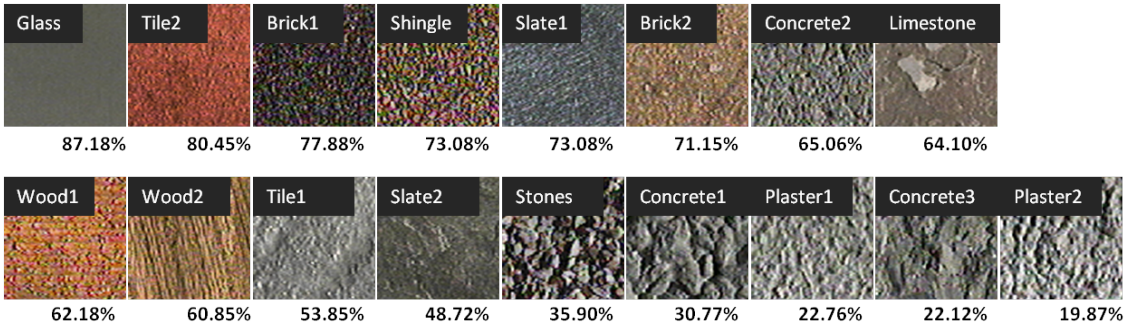


Figure 4.10: Individual classes classification accuracy on color images from Curet database.

Resolution	Optimal Bandwidth
6 in	0.9
4 in	0.9
2 in	0.7

Table 4.1: Optimal bandwidths chosen for classification for the images in each resolution. These values varied depending on the particular training regions chosen, so the values producing the best results on average were used.

performance of the texture features. As shown in Figure 4.11 the classification accuracy decreases for samples with higher roughness, probably due to the confusing 3D effects that occur in these classes. Neither methods of classifier, assuming a histogram model, or enforcing normal statistics adequately take these into account.

4.3 Purdue

The results of the previous sections of this chapter indicate that the simple method of incorporating color content via concatenation is nearly equivalent to the other more complex methods, thus it was used exclusively along with the Bhattacharyya distance to classify regions of the Purdue dataset. For each resolution, a bandwidth was chosen that optimized overall classification rates, but the optimal values did not differ significantly from each other. The values are shown in Table 4.1.

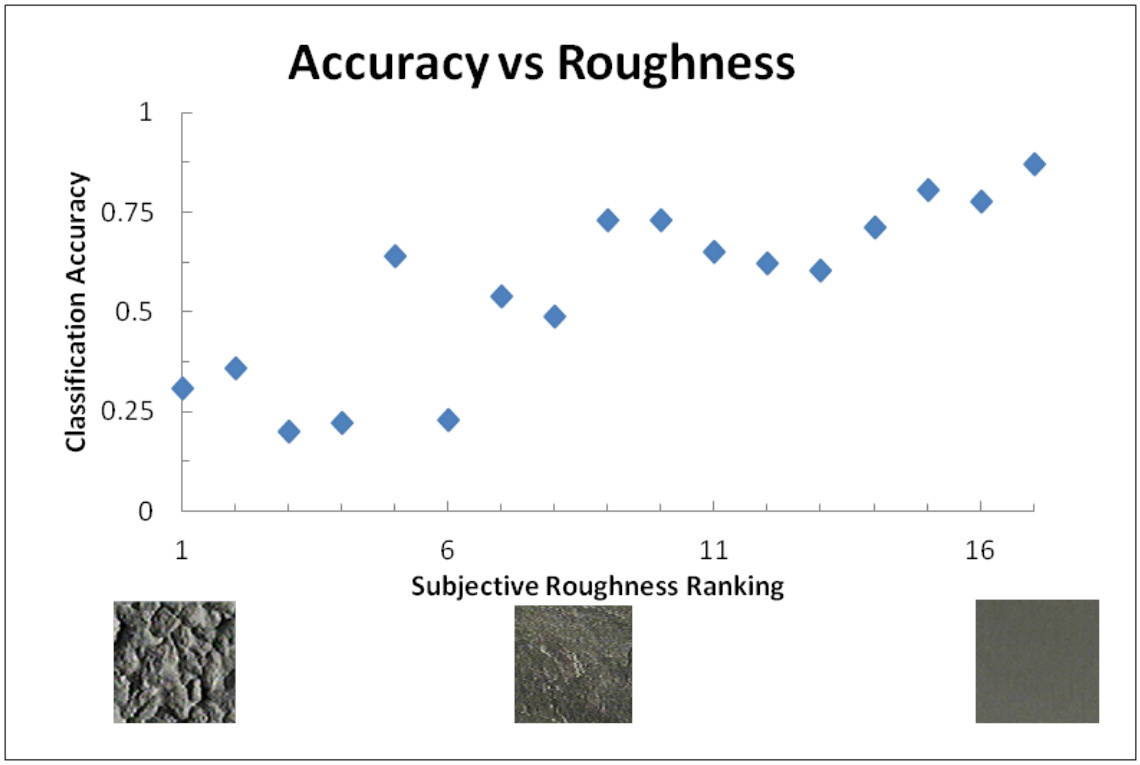


Figure 4.11: Classification accuracy versus subjective evaluation of class roughness ranking. Rougher classes exhibit higher 3D effects such as self shadowing and are more difficult to classify.

4.3.1 Classification Results - Overall Accuracy

First the overall accuracy of the six classes was investigated. In the following plots, the notation "X/X" refers to the cardinal direction the training data was taken from, followed by the direction the test data was taken from. Therefore the plot labeled E/N shows results using an East looking training image (or images) and North looking test data.

Figure 4.12 shows the overall accuracy for each resolution as the number of training regions taken from the input image increases. The difficulty in classification across images which differ in viewing direction can be seen, especially in the 4 in case. Here the overall accuracy varies by more than 10% as the cardinal direction changes. The 2 in case shows a higher average overall accuracy over all of the different look directions, and a much tighter spread in accuracy when classifying across different look angles. Overall, these results indicate that classification using a single input image (containing between 3 to 7 regions) can achieve accuracies of 65-70%. The tendency to overfit seems not to exist, as after the maximum number of training regions used the accuracy retains a slight upward trend in most cases. The exception to this, the "E/N" images from the 6 in resolution, show greater sensitivity to the choice of training data, indicating greater within class diversity in this feature space, and an inability of the classifier to successfully incorporate relevant data points and ignore outliers.

4.3.2 Classification Results - Class Breakdown

Class Accuracy

Figures 4.13 and 4.14 show how individual classes break down. On the whole, the "White Membrane (TPO)" and "Pk" classes perform best, greater than 90% in most cases, followed by the "ballast" roof class at nearly 70%. At every resolution and combination of training and testing data the "Rubber" and "Shingle" classes performed the worst, never exceeding 50% accuracy. The same relative performance of every class was observed over all resolutions, except that the "White Membrane (TPO)" class dropped significantly in accuracy at the 4 in resolution. Because in every other case the class trends remained the same, it is assumed that this is from the diversity of the training data, rather than an artifact of that particular resolution. In most cases the 2 in resolution performed the best, except that the "Rubber" and "Shingle" classes showed a slight advantage at the 6 in resolution when averaged over all look angles, as in Figure 4.14.

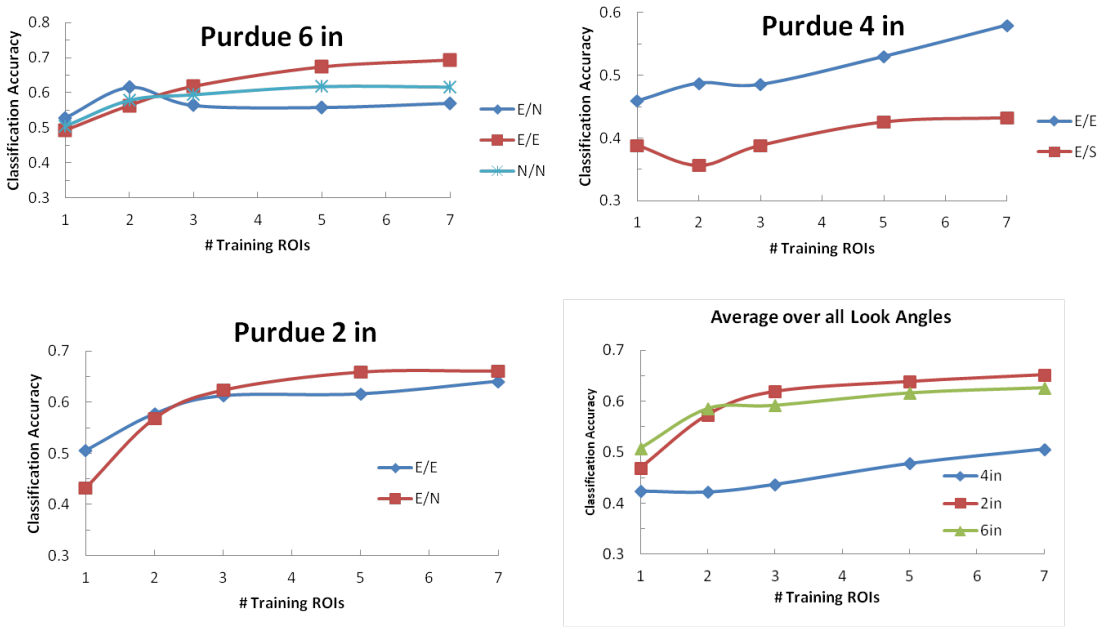


Figure 4.12: Classification accuracy versus number of training regions for each resolution and combination of training and testing image. Legend indicates cardinal direction of training image/cardinal direction of test image.

Table 4.2: "White Membrane (TPO)" class accuracies for varying configurations of training and test data. The notation *(E1/E4)* refers to training image "E1", paired with test image "E4", both from the East facing set of images.

# Training	E1/E4	E4/E1	E1/S1	E4/S1	S1/E1	S1/E4
1	0.583	0.000	0.000	0.031	0.000	0.722
2	0.528	0.167	0.000	0.188	0.000	0.833
3	0.556	0.500	0.000	0.188	0.000	0.833
5	0.556	0.750	0.000	0.156	0.000	0.861
7	0.500	0.750	0.000	0.125	0.000	0.889

In this same figure, the standard deviation over all look angles for each resolution is shown. The 6 in resolution shows the most consistent performance over the three look angles tested, varying the least in accuracy as training and testing look angle was varied. The "Ballast" and "Clay Tile" classes varied the least over the various look angles and performed consistently for all resolutions. Ignoring the 4 in results, the accuracy for each class changes only 5-10% for the east facing images as resolution is changed. This same trend is observed including the 4 in resolution, except for the large drops in accuracy in the "White Membrane (TPO)" class, and the "Shingle" class. This explains the large class standard deviations as resolution is changed in Figure 4.14 for those two classes.

A note on the 4 inch resolution results is in order given its poor performance. Figure 4.13 shows that the majority of the performance decrease for the 4 inch resolution appears to be coming from the decrease in individual class accuracy of the "White Membrane (TPO)" class. Referring to Table 4.2 There are 3 particular configurations of training and testing images which result in almost no successful classifications. These particular configurations coincide with limited class data. Table 3.2 shows that the "E1" and "S1" images only contain 3 and 4 representative "White Membrane (TPO)" classes. When these images are used as test cases the classification fails dramatically probably in part due to the limited number of examples. The upshot of this observation is simply that further studies with expanded datasets are necessary.

Regions

Examining the individual region successful and unsuccessful classifications it appears that the within class diversity of training data is responsible for individual class failures. Figures 4.15 and 4.16 show the training and test regions for a particular classification on the 6

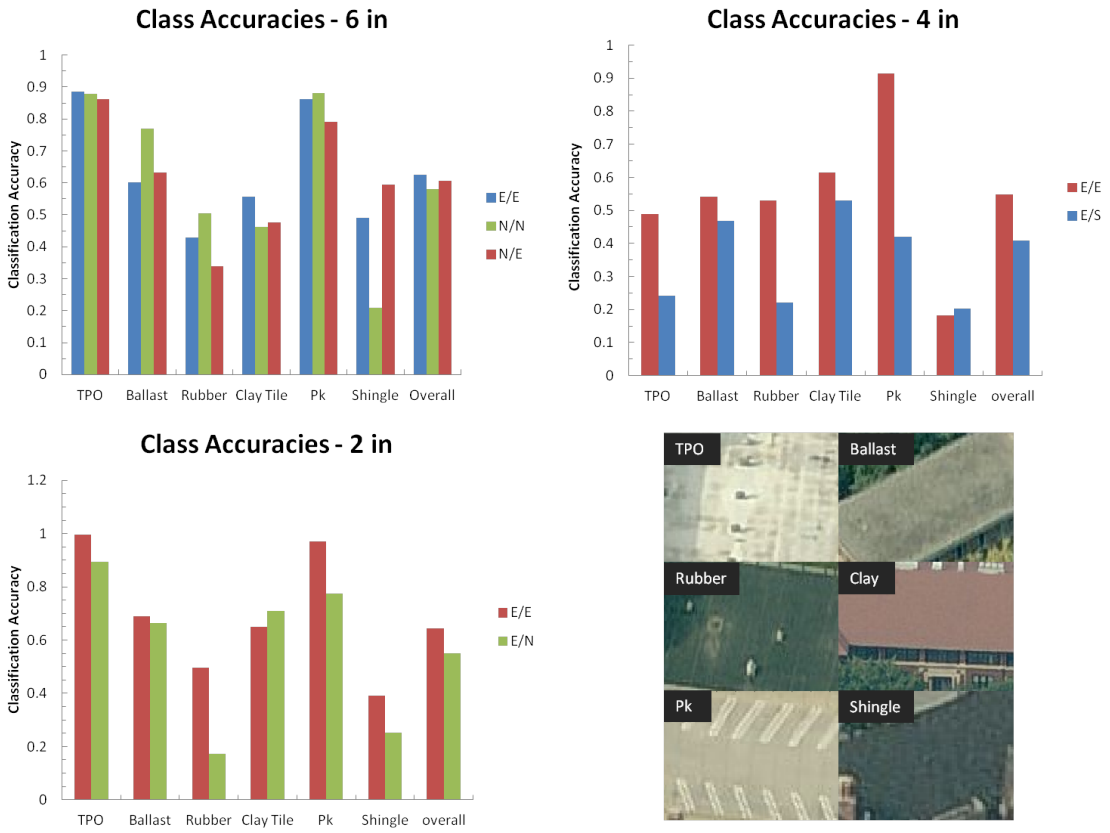


Figure 4.13: Classification accuracy of individual classes in each resolution and train/test image averaged over all training samples.

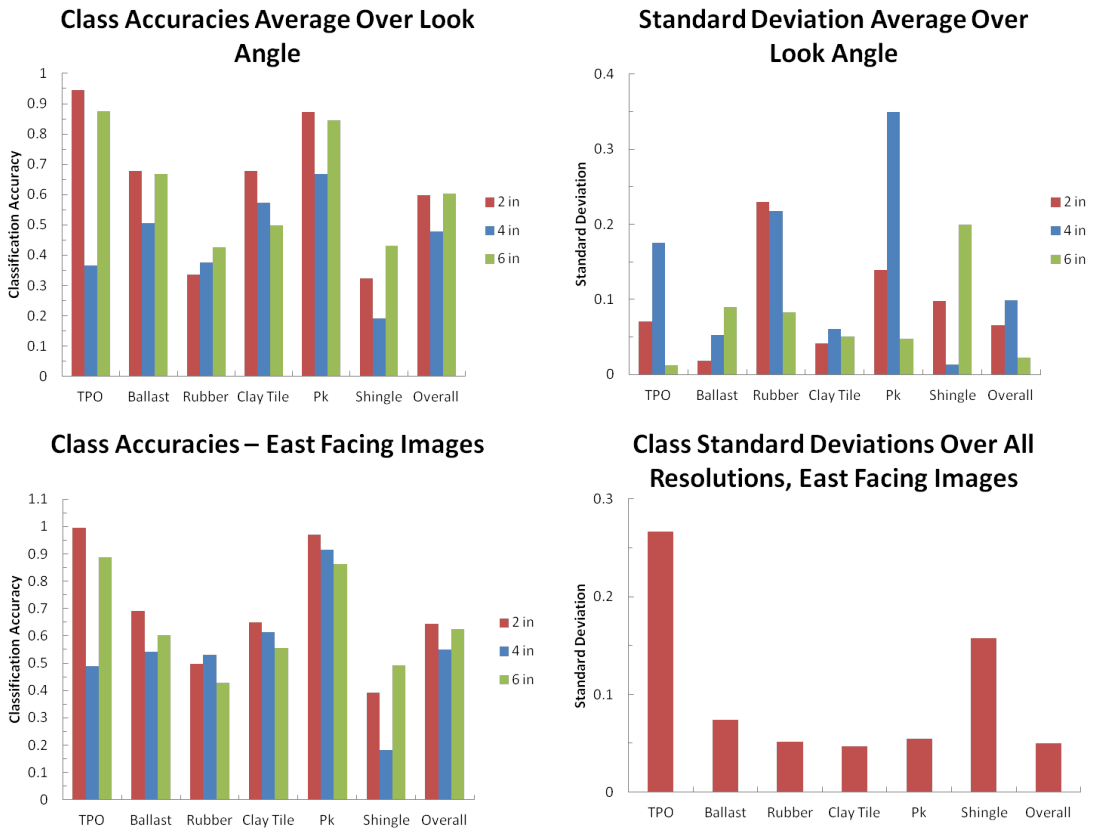


Figure 4.14: Classification accuracy of individual classes averaged over all look angles with standard deviations.

in resolution images. For the "Clay Tile" class, training regions were selected from a sunlit portion of the roof near the periphery of the image, and shaded roof portions were successfully classified in the test image.

Figure 4.17 show zoomed portions of other example classifications from the 6 in resolution. It is shown that the classifier successfully labels a rotated "Pk" class correctly, but that an error is made classifying the less textured "Rubber" region despite being trained on a nearly identical patch. Figure 4.18 shows a "Clay Tile" region this time trained on a shaded roof portion, successfully classifying the shaded test regions despite the additional roof structures present, but still failing on the opposite portions. It is also observed that certain "Shingle" regions of a different color are correctly labeled, but an error is made on a similarly textured region with a lighter color.

Figure 4.20 shows class regions from the 2 in resolution. Again, the classifier seems to easily handle variations in overall contrast level, correctly classifying the shaded "Clay Tile" roof portions, but cannot handle the diversity of the "Shingle" class entirely, as it fails in Figure 4.20.

4.3.3 Class Distributions and Feature Images

To further examine the classification results, and some of the reasons behind these the class distributions were investigated. Figure 4.21 shows the marginal class distributions for each class over several particular filters at the 6 in resolution. For the first six filters, which occupy the lowest frequencies, there is inadequate class separability, but in the mid range frequencies, filters 13-18, much greater separability is attained. This shows clearly that certain frequency channels are useless in the separation of these particular classes, and an approach that optimized frequency bands with respect to separability would improve results.

The individual class spreads show that the "Shingle" class, which performed second worst at this resolution varies the most widely in its response to these filters. It is also apparent that it possesses two strong peak responses, indicating two distinct clusters. The best performing classes, "White Membrane (TPO)" and "Pk", are the most separated from the other distributions. The most commonly confused classes, "Shingle" and "Rubber" overlap significantly. In general the distributions do not appear to be Gaussian.

Figure 4.22 shows on each plot a single filter response from one class at all resolutions. The "White Membrane (TPO)" class distributions are largely similar, with the

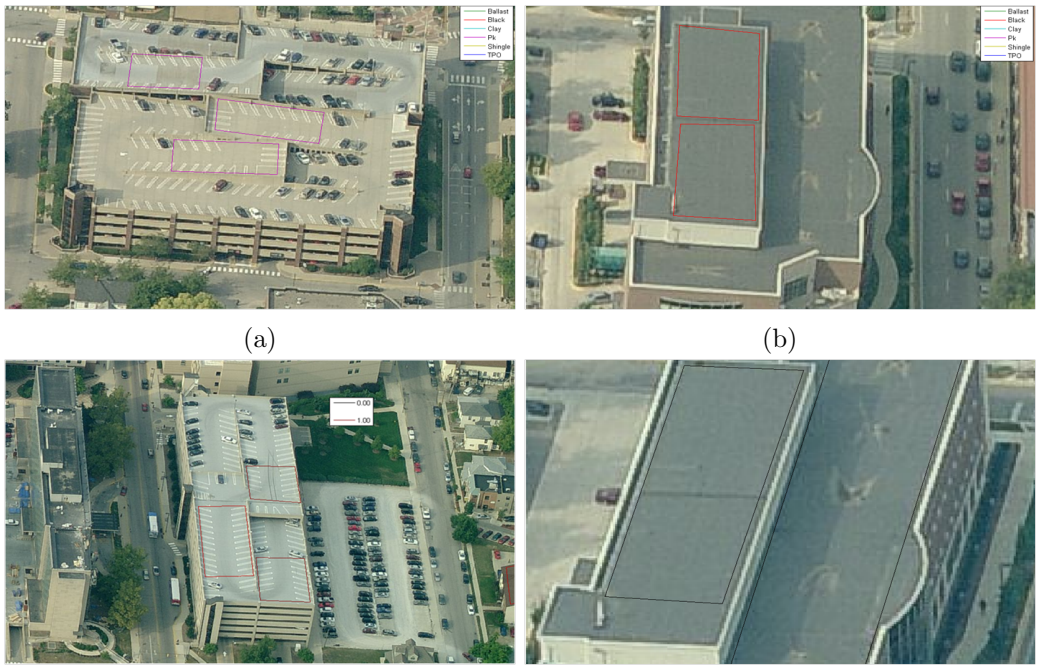
Figure 4.15: Example of training regions selected from a 6 in resolution image.

Figure 4.16: Classification results from a 6 in resolution image. The legend colors indicate successful or unsuccessful classification.

peak responses tending toward slightly lower values as resolution increases in 4.22a. In contrast, the "Shingle" distributions over the frequency ranges shown have a very high spread, and show at least two distinct peaks at the 4 and 2 in resolutions. The spread at the 6 in resolution is about the same as the 2 in, but the 2 in resolution is able to further distinguish the peak splitting. The opposite phenomenon occurs for the "Ballast" class where the spread appears to tighten up as the resolution increases.

Figure 4.23 shows the origin of the class distributions shown earlier. Immediately apparent are the edge effects that arise inherently from application of a filter with a wide region of support. Unfortunately these are responsible for a large part of the class distribution. The valuable part of the distribution has a much smaller spread, and shows much less correlation than the distribution as a whole.

Figure 4.24 shows how the overall class distribution is built up, and the resulting within class diversity. In every case the distributions show a large quantity of outliers due to edge effects.



(a) (b)

(c) (d)

Figure 4.17: Training (a-b), and testing (c-d) regions showing successful classification of varying orientation "Pk" class, and unsuccessful classification of "Rubber" class, at 6 in resolution. Line colors indicate classification accuracy; black - misclassified region, red - correctly classified region.

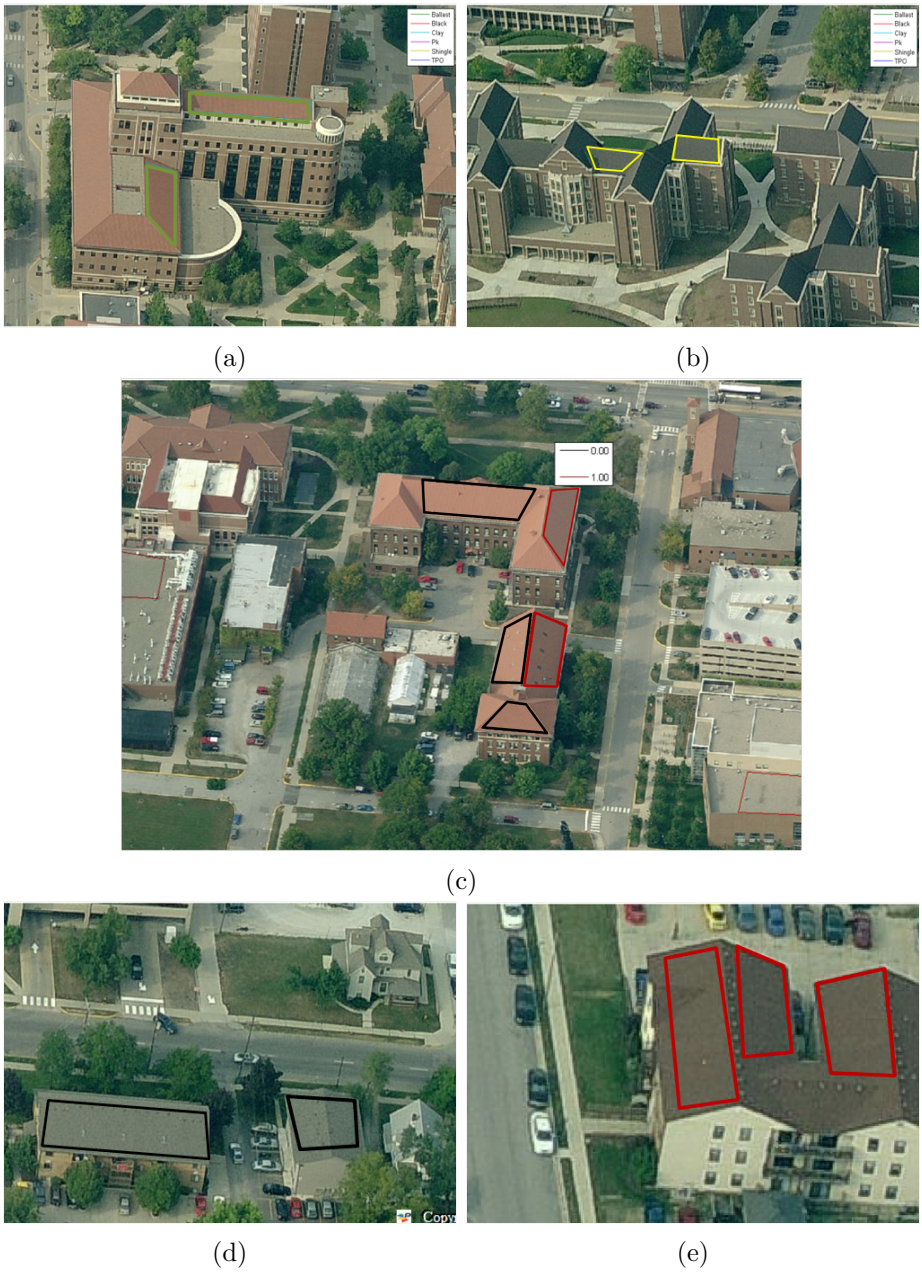


Figure 4.18: Further region by region classification results from 6 in resolution. Training regions from the (a) "Clay" class, and (b) "Shingle" class, and (c-e) test regions. In (c-e) line colors indicate classification accuracy; black - misclassified region, red - correctly classified region. Classification was successful on the west facing "Clay" roofs, but not the east, though the "Clay" training region was taken from the east direction. (e) shows a successful classification on a perpendicularly oriented "Shingle" region, but an unsuccessful classification in (d).

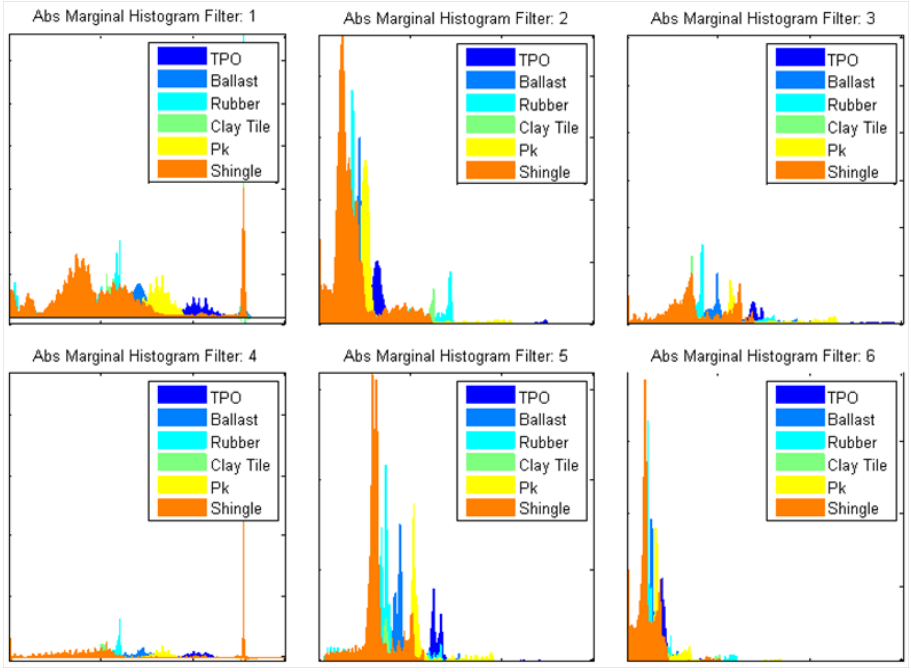
Figure 4.19: Training data selected from a particular 2 in image

(a)

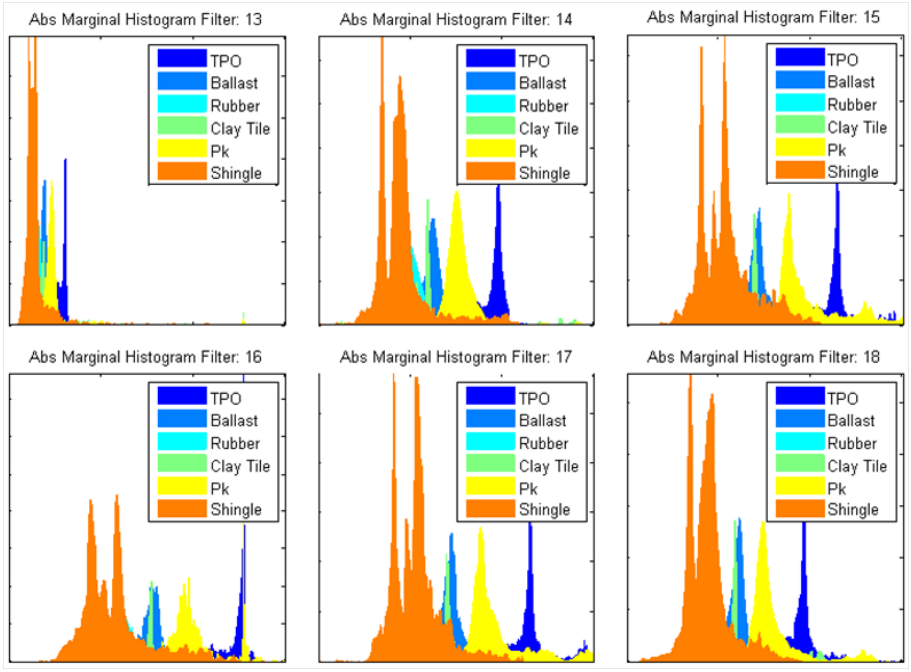
(b)

(c)

Figure 4.20: Test results from the 2 in image shown in Figure 4.19. (a) shows shaded and sunlit portions of the "Clay" region classified correctly. (c) shows "Shingle" regions oriented slightly differently from the training regions classified successfully in only 3 of 5 cases. Line colors indicate classification accuracy; black - misclassified region, red - correctly classified region.



(a)



(b)

Figure 4.21: Marginal class distributions from filters 1-6 (a), low frequency, and 13-18 (b), middle frequencies. The higher center frequency filters lead to greater separation.

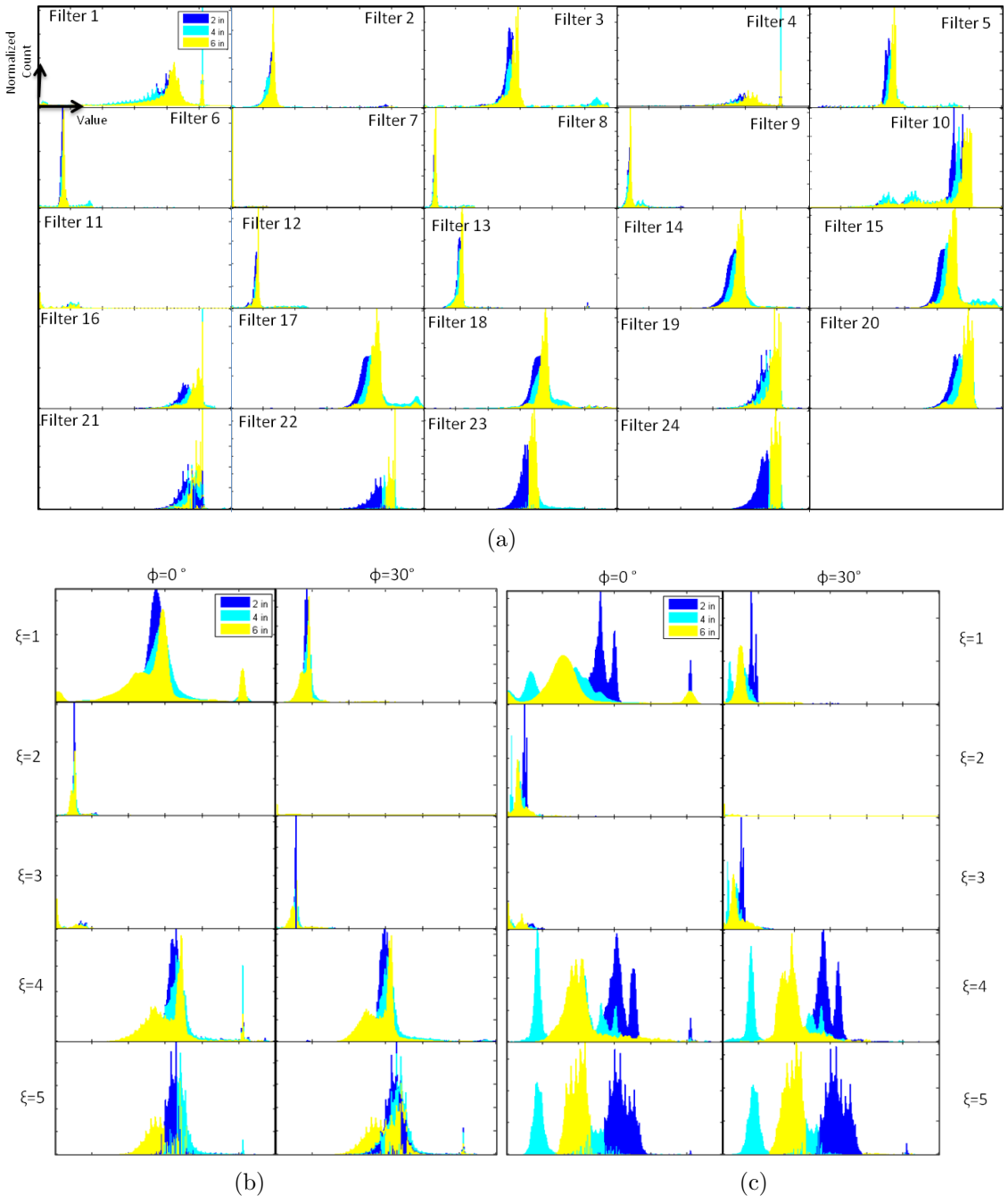


Figure 4.22: Marginal class distributions over a range of filters. 4.22a "White Membrane (TPO)" class and filters 1-24, 4.22b-4.22c "Ballast" and "Shingle" classes, 10 filters. All three resolutions shown on each plot.ϕ and ξ are the orientation and center frequency index respectively.

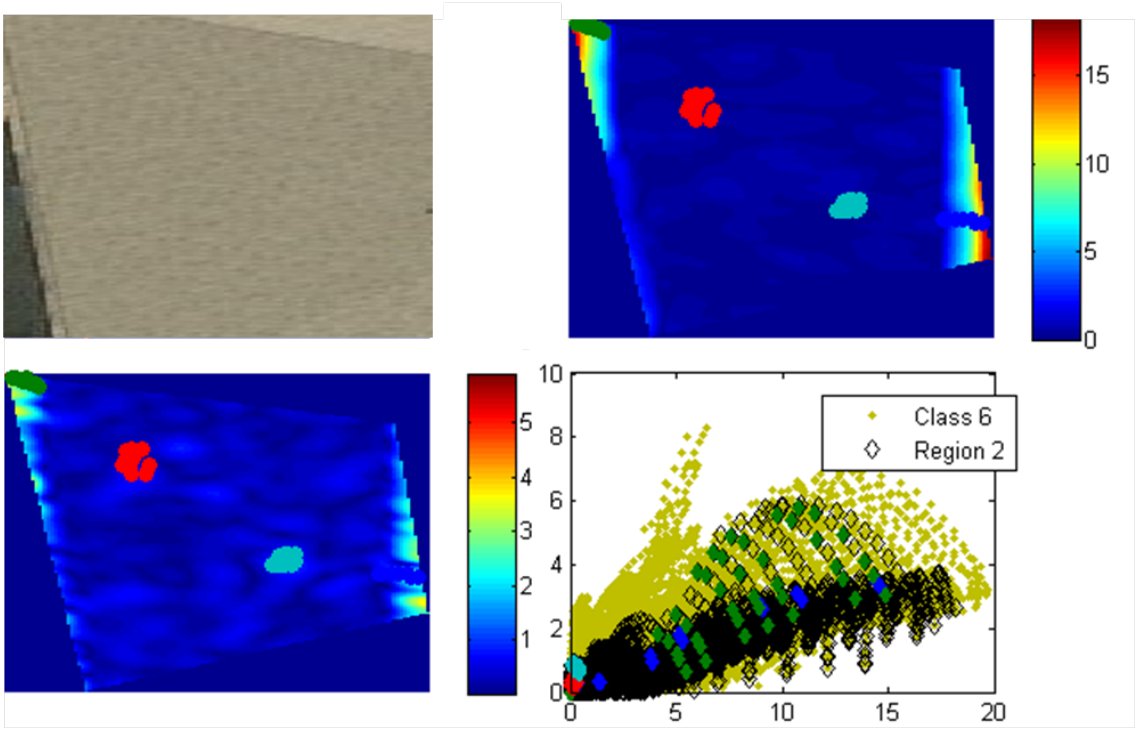


Figure 4.23: Input image from "Shingle" class (top left) and its filter responses from two filter dimensions. Plot shows response of selected pixels from the images plotted on top of the "Shingle" overall class distribution, and the selected region's class distribution in the two dimensions shown.

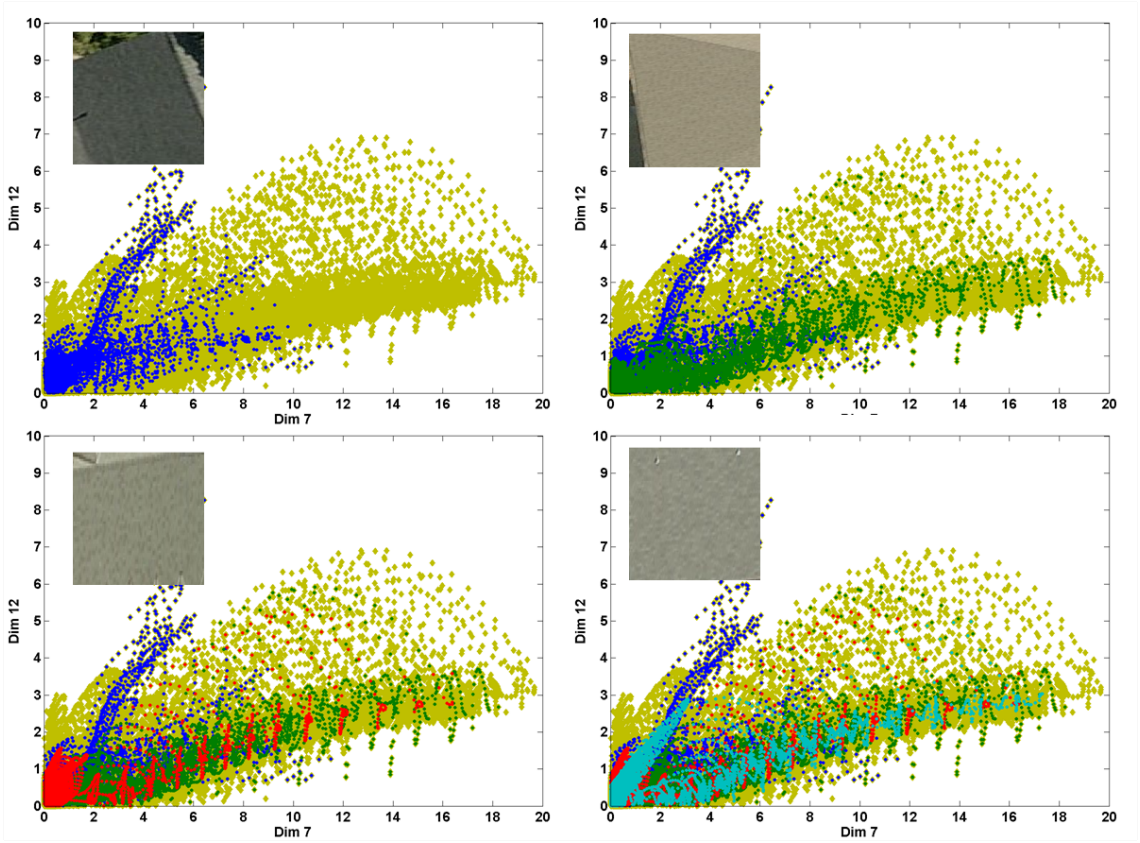


Figure 4.24: Overall "Shingle" class distribution plotted with successively more of its region's entries highlighted, showing the within class variation. Much of the data unfortunately is generated by edge effects, and are outliers.

Chapter 5

Conclusions

5.1 Conclusions

We have investigated the performance of a standard filter based classifier in the labeling of roof materials in oblique imagery. The unique dataset used provided imagery at multiple GSD's and camera look angles and presented major challenges to the classifier including projective distortions, scale changes, and illumination differences. In addition to these variables there were also substantial within class variations among the six roof classes chosen. The major inquiries investigated consisted of the role of color and texture in the classification task, which among several proposed methods was the best for incorporating color information into the classifier, how much training data was necessary to adequately classify images from varying look angles, and whether or not a standard filter based classifier could deal with the aforementioned variables in image acquisition.

5.2 Initial Tests

The tests of rotation invariance, and combining color and texture gave insight into important elements of the filter based classifier.

5.2.1 Brodatz Tests

First, it was a simple matter to incorporate rotation invariance into this classifier, due to its desirable properties of periodicity along the orientation dimension, and the shift

invariance property of the DFT. The results suggested that this method provided a good incorporation of rotation invariance, classifying half of the 30 Brodatz classes with perfect accuracy, and most of the rest with near perfect accuracy (only 1 misclassification per class). The results were moderately sensitive to the bandwidth as a linear increase in accuracy was noted with increasing bandwidth.This is sensible, since a larger bandwidth filter will capture more frequencies in the same channel, further aligning the dominant spectral components of the input and rotated images. In a real world situation, where the rotated images are not identical in every other way, the larger bandwidths may lead to a decrease in accuracy due to the increased overlap. Therefore in practice the accuracies probably would not reach the levels shown in Figure 4.1.

5.2.2 Color Tests

The results on the Curet database indicate that the histogram method was not suitable for this task, in the implementation that was done here. In contrast to the Bhattacharyya method, the histogram method does not impose a statistical model of the data, which is beneficial, however, each bin of the resulting histogram is given equal weight, whereas specifying a Gaussian distribution essentially emphasizes the center of the distribution. The distance metric in the histogram case is thus more sensitive to outliers than using the Bhattacharyya distance. This may be one reason for the failure of the histogram method.

The results of Section 4.2 also indicate that compared to the other methods, it is just as effective to concatenate the color channel intensities onto the filter response vector to form an augmented feature vector. This may not be an effective fusion method in every case, depending on the magnitude of the filter response and color data. This method also implies that the same classifier is used on data generated by separate processes, one by filtering for dominant texture components, and the other from color intensities. The method of voting may likely have performed better on the Purdue dataset, since two of the classes showed distinct differences in color. Incorporating a set of weights for each color channel or using a different classifier on the color and texture feature vectors would probably improve results.

5.3 Purdue Tests

The final results of classification on the Purdue dataset showed that this classifier, though not specifically accounting for the scale or projective distortions, was to some extent capable of handling them. Figure 3.6 shows the effects of these on the frequency spectra of some of the input images. The higher frequency Gabor filters also maintained a higher angular frequency, providing some flexibility in the case of changing the orientation of a dominant texture component such as these. Therefore, some robustness to projective distortions is gained. The results of Section 4.3 show that this is indeed the case. Many of the input training regions differed in orientation from the test images, and were successfully classified. The case of scale variation is similarly handled, since the Gabor filters are circularly symmetric, a dominant texture component which changes in scale may still be partially captured by the same filter.

The number of training regions necessary to characterize the class distribution appears to be only several. This means that one input image should be sufficient to classify the rest. The issue in constructing class distributions is that there is substantial within class variation including many outliers, so imposing a simple statistical assumption such as Gaussian data is incorrect. The figures in Section 4.3.3 indicate as well that there exist multiple subclasses within some of the overall roof classes (especially the "Shingle" class), shown as multiple peaks in the class distributions. A better way of assembling the training data is thus required. Until this is implemented, there will most likely be significant variation in classification accuracy from image to image.

On a per class basis the classes with the most obvious color and/or texture features were most successfully classified at all resolutions and look angles. These were the "Pk", "Clay", and "TPO" classes, which exhibited either strong distinct textural features, or distinct color components. It was shown in the figures of Section 4.3.3 that the "Shingle" class contained a very broad class distribution, and was often misclassified, as well as the "Rubber" class, these being very similar both spectrally, and in texture. The other classes in many cases could be classified with over 90% accuracy. The large difference in individual class accuracies is important to note in the overall performance of the classifier, and in assessing whether it is appropriate to use or not.

5.4 Conclusions

The aim of this work was to present and examine a classifier capable of accurately identifying material classes in aerial imagery with minimal training data for the application of aiding in generating synthetic imagery. The results and conclusions regarding the classifier's success indicate that for this application it has not achieved "roadworthiness" but is a promising step forward in this area. The main requirements of a classifier suitable for use in completely automated synthetic imagery generation were high accuracy, insensitivity to diverse training and test data (including sensor look angle, illumination), with minimal training data. The accuracies regularly achieved did not exceed 70%, but were mostly consistent for the classes studied even across varying ground sample distance. If the classifier were used immediately with no further modification, initial studies would be necessary to identify classes which were correct with high probability to reduce the need for a man in the loop. Otherwise, this level of accuracy would probably not be sufficient for most uses. To achieve the highest accuracies, the classifier required a minimal amount of training data, equivalent to the number of class examples found in one image, which is probably suitable in most cases. This indicates that the initial overhead involved in using this classifier for the purpose of synthetic scene generation is low. The feature set itself was shown to be somewhat resilient to changes in sensor look angle, rotations, and illumination differences, but at least one of the classes examined ("Shingle") showed subclasses in its overall class distribution. The classifier used inherently assumes that this is not the case, and cannot draw more complicated decision boundaries required for diverse training data. The classifier also does not indicate a level of confidence in its label, being only a hard classifier. A feature like this is necessary for this application, as any cross checking of the results should be confined only to low probability region labels to minimize operator intervention. Overall, the classifier investigated should be seen as a step forward toward the goal of automating scene generation in satisfying some of the requirements including low training data, and some robustness to the scene conditions, but leaves plenty of room for improvement in its accuracy, consistency, and ability to cope with varied scene content.

Chapter 6

Future Work

The classification results of the previous chapters show that the classifier as implemented show some robustness to the variables presented by the imaging conditions, but leaves much room for improvement, especially a more principled data fusion methodology. As well, it is imperative to have a more reliable method of assessing the confidence level of a particular classification so that the end user does not need to peruse the entire classified dataset. The method of data aggregation should also be investigated. Future improvements could be concentrated along the following lines.

6.1 Classifier

The method of aggregating the data in this work took feature images and the feature vectors therein and classified the feature image as a whole using the Bhattacharyya distance. This provided some robustness against clutter and noise in the image, as well as additional rotation invariance, but the drawback is that it provided a coarse classmap, and an ineffective confidence metric based on a distance metric rather than a probability. Classifying each pixel would lead to a much higher fidelity classmap which could then be aggregated by weighting the confidence of each pixel's class assignment, perhaps leading to a better overall region assignment, or a segmentation into additional regions. Figure 6.1 shows the difference in these two methodologies. Initial tests using the Mahalanobis distance classifier (to compare more directly against the Bhattacharyya minimum distance classifier) show that even a simple method of aggregating the data following classification rather than prior, shows better results. These tests were performed on the 6 in Purdue

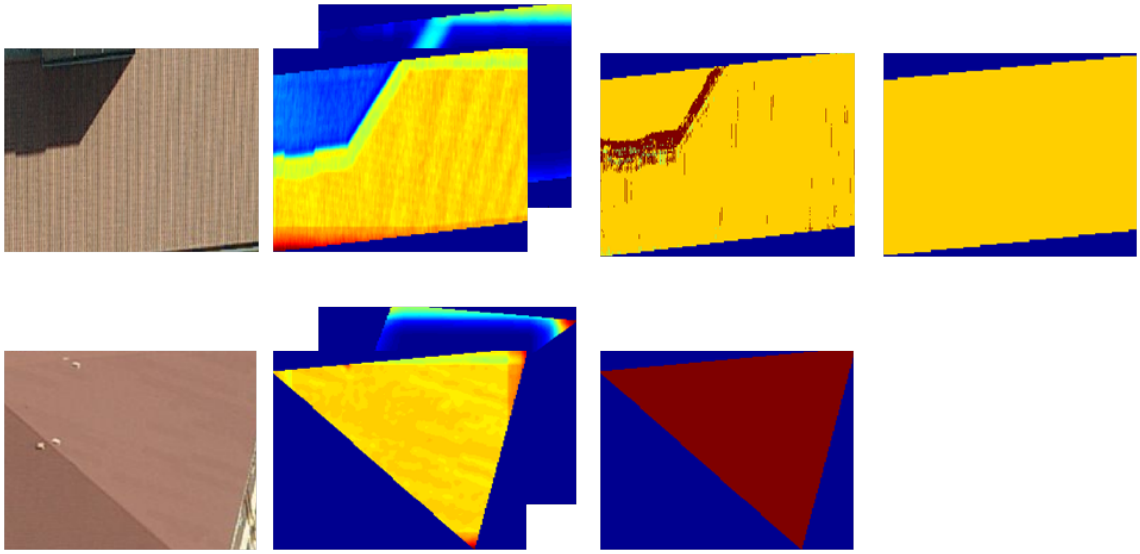


Figure 6.1: Two possible workflows for the classification process. (top) shows the filter responses classified such that each pixel's feature vector is classified, then the entire region's classmap is aggregated to give the region a label. In the bottom track the region's feature images are classified entirely, with each pixel's feature vector put into a distribution and then classified.

dataset classifying a test image given the regions of one training image. Figure 6.2 shows these results compared to the previous results using the Bhattacharyya distance.

The Bhattacharyya distance is a simple classifier, and presumes normally distributed data. Because of high within class variation these do not accurately describe the data. A better method for combining training regions into a class distribution, such as using textons as in [37] would allow only the parts of each class which coincide to be used as training data. Another option is to use a non-parametric classifier capable of forming more complex class boundaries, however this may lead to overfitting in the presence of outliers.

Additionally, it would be desirable to automatically select the appropriate bandwidth, and even the particular filters used given a database of training classes. This can be done by maximizing the classification accuracy using some subset of the training data as "test" data. Automatically selecting the bandwidth in this way can be done offline, and the optimization of particular filters used reduces the size of the feature space speeding up computation time. These improvements make the implementation of this classifier more feasible.

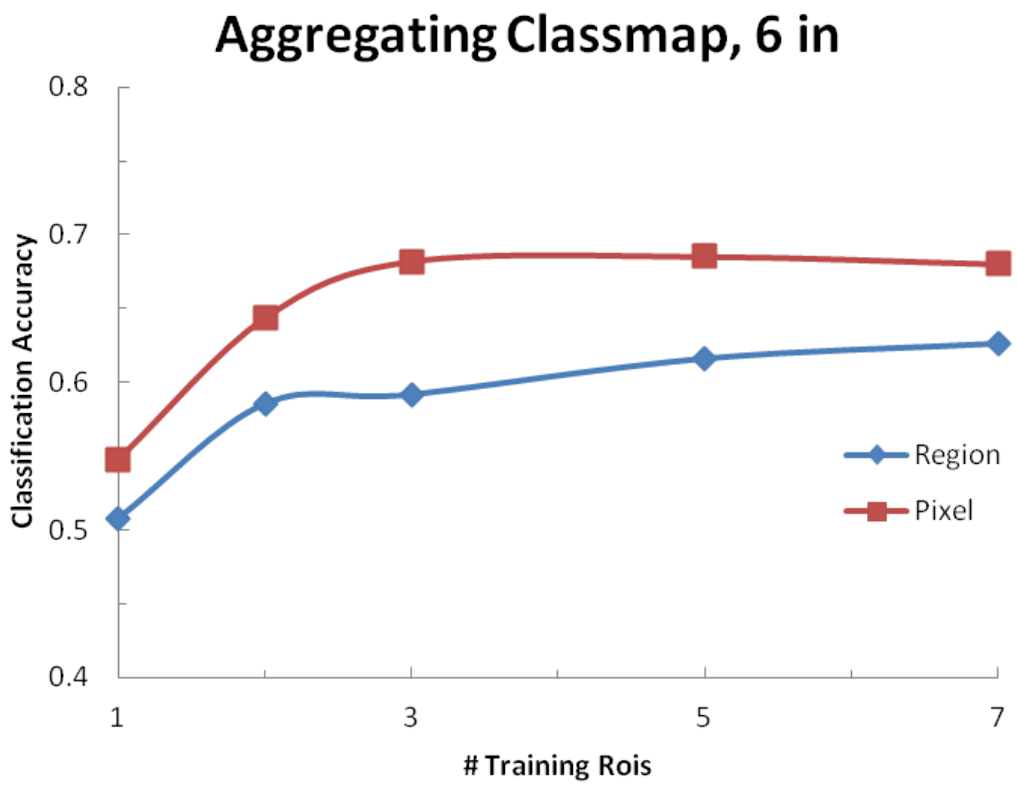


Figure 6.2: Comparison of the two methods of data aggregation shown in Figure 6.1. The method which classifies individual pixels first, then aggregates the resulting classmap to label the region outperforms the other method, which was used in the experiments shown previously.

6.2 Confidence Metric

An indicator of classification success is desirable; if the end user must comb the entire dataset to validate the classifier's success this defeats the purpose of a supervised classifier. The initial attempt to incorporate a confidence metric using the Bhattacharyya distances output for each region was tested by classifying several regions of the Purdue dataset, and thresholding the output data, keeping only the high confidence points. These points were then assessed for their accuracy. The threshold was increased such that successively more and more lower confidence data was thrown away, until only the high confidence data remained. Ideally, the confidence metric would pinpoint only misclassified points, and so the accuracy would continue to increase with every point thrown away. This was not the case for the confidence metric based on the Bhattacharyya distance, as shown in Figure 6.3. Improvements could easily be made here by using a soft classifier, or one based on Bayesian probabilities to provide a more accurate assessment of confidence.

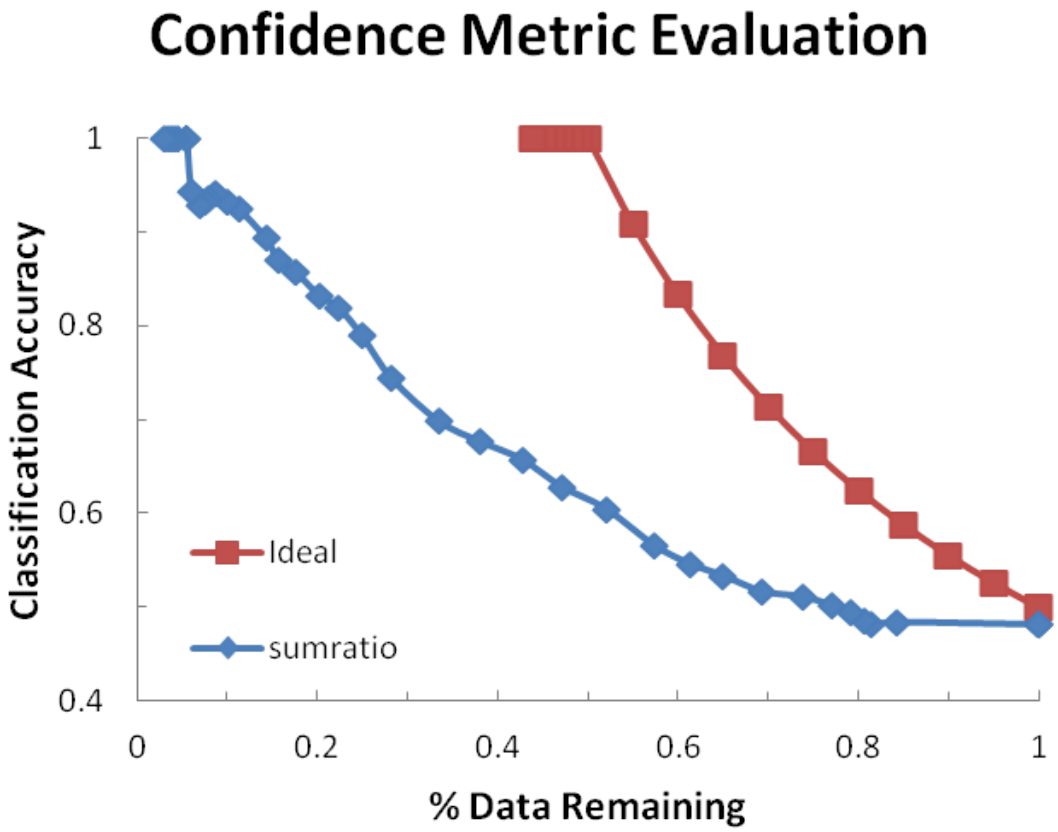


Figure 6.3: Evaluation of a confidence metric based on the ratio of Bhattacharyya distances of each test pixel to each class. In the ideal case, as subsequently more lower confidence data was thrown out, the remaining data should be accurately classified. The confidence metric used here is not as effective, only retaining exclusively correctly classified data after the bottom 90% of classified data was thrown out.

Bibliography

[1] E. J. Ientilucci and S. D. Brown, "Advances in wide area hyperspectral image simulation," in *Conference on Target and Backgrounds IX*, vol. 5075, 2003, pp. 110–121.

[2] D. Nilosek and C. Salvaggio, "Applying computer vision techniques to perform semi-automated analytical photogrammetry," *IEEE Xplore*, pp. 1–5, 2010.

[3] D. Nilosek, S. Sun, and C. Salvaggio, "Geo-accurate model extraction from three-dimensional image-derived point clouds," in *Proceedings of the SPIE, SPIE Defense and Security Sensing, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVIII, Modeling and Simulation.*

[4] A. Jain and M. Tuceryan, *Texture Analysis.* Singapore: World Scientific, 1993.

[5] D. A. Clausi and T. E. Jernigan, "Designing gabor filters for optimal texture separability," *Pattern Recognition*, vol. 33, no. 11, pp. 1835–1849, 2000.

[6] T. Randen and J. H. Husoy, "Filtering for texture classification: A comparative study," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 291–310, 1999.

[7] C. Schmid, "Constructing models for content-based image retrieval," in *Conference on Computer Vision and Pattern Recognition.* Ieee Computer Soc, 2001, pp. 39–45.

[8] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.

[9] M. Varma and A. Zisserman, "Classifying images of materials: Achieving viewpoint

and illumination independence," *Computer Vision - Eccv 2002 Pt Iii*, vol. 2352, pp. 255–271, 2002.

[10] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[11] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer Vision Graphics and Image Processing*, vol. 29, no. 1, pp. 100–132, 1985.

[12] D. G. Stork and H. R. Wilson, "Do gabor functions provide appropriate descriptions of visual cortical receptive-fields," *Journal of the Optical Society of America a-Optics Image Science and Vision*, vol. 7, no. 8, pp. 1362–1373, 1990.

[13] D. Gabor, "Theory of communication," *Journal of the institute of electrical engineers*, vol. 93, pp. 429–457, 1946.

[14] M. R. Turner, "Texture-discrimination by gabor functions," *Biological Cybernetics*, vol. 55, no. 2-3, pp. 71–82, 1986.

[15] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[16] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 55–73, 1990.

[17] N. P. Angelo and V. Haertel, "On the application of gabor filtering in supervised image classification," *International Journal of Remote Sensing*, vol. 24, no. 10, pp. 2167–2189, 2003.

[18] C. J. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *Ieee Transactions on Image Processing*, vol. 11, no. 4, pp. 467–476, 2002.

[19] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the optical society of America-A*, vol. 2, no. 7, pp. 1160–1169, 1985.

[20] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. USA: Elsevier, 2009.

[21] CIE, "Colorimetry," CIE, Tech. Rep., 2004.

[22] F. Bianconi, A. Fernandez, E. Gonzalez, D. Caride, and A. Calvino, "Rotation-invariant colour texture classification through multilayer ccr," *Pattern Recognition Letters*, vol. 30, no. 8, pp. 765–771, 2009.

[23] F. Bianconi, R. Harvey, P. Southam, and A. Fernandez, "Theoretical and experimental comparison of different approaches for color texture classification," *Journal of Electronic Imaging*, vol. 20, no. 4, p. 17, 2011.

[24] T. Maenpaa and M. Pietikainen, "Classification with color and texture: jointly or separately?" *Pattern Recognition*, vol. 37, no. 8, pp. 1629–1640, 2004.

[25] A. Drimbarean and P. F. Whelan, "Experiments in colour texture analysis," *Pattern Recognition Letters*, vol. 22, no. 10, pp. 1161–1167, 2001.

[26] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66–75, 1994.

[27] T. Randen and J. H. Husoy, "Multichannel filtering for image texture segmentation," *Optical Engineering*, vol. 33, no. 8, pp. 2617–2625, 1994.

[28] J. Yang, "Do gabor functions provide appropriate descriptions of visual cortical receptive-fields - comment," *Journal of the Optical Society of America a-Optics Image Science and Vision*, vol. 9, no. 2, pp. 334–336, 1992.

[29] J. Movellan, "Tutorial on gabor filters," University of California, Tech. Rep., 2005.

[30] T. N. Tan, "Rotation invariant texture features and their use in automatic script identification," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 751–756, 1998.

[31] F. Lahajnar and S. Kovacic, "Rotation-invariant texture classification," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1151–1161, 2003.

[32] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, "Kernel density estimation via diffusion," *Annals of Statistics*, vol. 38, no. 5, pp. 2916–2957, 2010.

[33] P. Brodatz, *Textures: A photographic album for artists and designer.* New York, NY: Dover, 1966.

[34] K. J. Dana, B. Van Ginneken, S. K. Nayar, and J. J. Koenderink, "Reflectance and texture of real-world surfaces," *Acm Transactions on Graphics*, vol. 18, no. 1, pp. 1–34, 1999.

[35] T. Randen, "Brodatz textures." [Online]. Available: http://www.ux.uis.no/ tranden/

[36] M. Varma and A. Zisserman, "Texture classification," 2007.

[37] ——, "A statistical approach to texture classification from single images," *International Journal of Computer Vision*, vol. 62, no. 1-2, pp. 61–81, 2005.