

# A Compression Based Distance Measure for Texture

Bilson J. L. Campana

Eamonn J. Keogh

University of California, Riverside

{bcmpana, eamonn}@cs.ucr.edu

## ABSTRACT

The analysis of texture is an important subroutine in application areas as diverse as biology, medicine, robotics, and forensic science. While the last three decades have seen extensive research in algorithms to measure texture similarity, almost all existing methods require the careful setting of many parameters. There are many problems associated with a surfeit of parameters, the most obvious of which is that with many parameters to fit, it is exceptionally difficult to avoid over fitting. In this work we propose to extend recent advances in Kolmogorov complexity-based similarity measures to texture matching problems. These Kolmogorov based methods have been shown to be very useful in intrinsically discrete domains such as DNA, protein sequences, MIDI music and natural languages; however, they are not well defined for real-valued data. Towards this, we introduce the Campana-Keogh (CK) video compression based method for texture measures. These measures utilize state-of-the-art video compressors to approximate the Kolmogorov complexity. Using the CK method, we create an efficient and robust parameter-free texture similarity measure, the CK-1 distance measure. We demonstrate the utility of our measure with an extensive empirical evaluation on real-world case studies drawn from nematology, arachnology, entomology, medicine, forensics, ecology, and several well known texture analysis benchmarks.

## Keywords

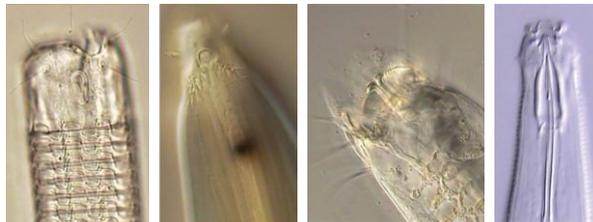
Image Similarity, Classification, Clustering, Video Compression

## 1. INTRODUCTION

Texture analysis is used in classification, clustering, segmentation and anomaly detection in images culled from domains as diverse as biology, medicine, robotics, biometrics, forensic science, and the study of historical texts. Texture recognition systems can have surprising uses; for example in Malaysia, a leading exporter of hardwoods, texture recognition is used to check against the logging of protected wood species and against attempts to pass off inferior strength species as stronger wood species for strength critical applications [23].

In the Content-Based Information Retrieval (CBIR) community, there has been extensive research in algorithms to measure texture similarity; however virtually all existing methods require the careful setting of many domain-specific parameters. For example, the commonly used Gabor filter requires the setting of scales, orientations, and filter mask size parameters [39][42]. As researchers have recently noted, “*Gabor filters show a strong dependence on a certain number of parameters, the values of which may significantly affect the outcome of the classification procedures*” [3].

Of the many problems associated with an abundance of parameters, the most obvious is simply that with many parameters to fit, it is exceptionally difficult to avoid over fitting [13]. An additional problem of parameter-laden algorithms is that they make it exceptionally difficult to reproduce published experimental results and to truly understand the contribution of a proposed algorithm [16].



**Figure 1: Examples of nematode diversity as seen under magnification**

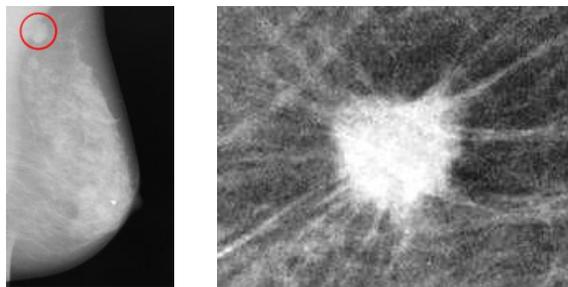
In this work we propose to extend recent advances in Kolmogorov complexity-based similarity measures [10][16][28][29] to texture matching problems. These Kolmogorov based methods have been shown to be very useful in intrinsically *discrete* domains such as DNA, natural languages, protein sequences, and symbolic music sequences such as MIDI or Parsons code; however, they are not defined for *real*-valued data such as textures. We show that by approximating the Kolmogorov complexity with the Campana-Keogh (CK) method of using state-of-the-art video compressors, such as MPEG, we can create an efficient and robust texture similarity measure. To give our ideas a concrete grounding, we will discuss in detail two motivating examples.

**Nematodes** are a diverse phylum of “wormlike” animals, and one of the most diverse of all animal groups. Nematode species are very difficult to distinguish; over 80,000 have been described, however the true number may be closer to 500,000. As shown in Figure 1, nematode bodies are semi-transparent structures which mostly consist of digested foods and fat cells.

Understanding the biodiversity of nematodes is critical for several applications such as pest control, human health, and agriculture. For example, millions of people are infected by nematodes worldwide with a quarter of the world’s population infected by a single genus of nematodes, *Ascaris* [2].

Because of their diversity and abundance, finding distinct characteristics of a nematode species for classification is a non-trivial task. Identification by experts requires three to five days to accomplish [14]. While the shape of the head and tail can be a useful feature in some cases, it is not enough to distinguish down to even the genus level. However, as we can see in Figure 1, nematodes are often richly textured, both externally and (given that they are semi-transparent) internally. As we shall show, our simple texture measure based on the CK method is extremely effective in classifying nematodes, without the need for careful parameter tuning or human-guided feature extraction.

**Breast cancer** results in about 500,000 deaths each year [17]. The survival rate of breast cancer patients greatly depends on an early diagnosis. In the US, survival rates of early diagnosed patients are 98%, where the survival rate of a regionally spread cancer is 84%, and those in a late stage where distant organs are effected have a survival rate of 28% [21]. Figure 2 displays an annotated image from the Mammographic Image Analysis Society mammogram database [44] with a malignant mass inscribed.



**Figure 2: left) A mammogram image with a malignant mass encircled. right) Cancerous lesions tend to invade the surround tissue and exhibit a radiating pattern of linear spicules, resulting in unusual textures**

Numerous trials and evaluations have shown that mammography is the single most effective method for

early detection of breast cancer and greatly increases chances of survival and treatment options[15][20][45]. Radiologists analyze mammograms for the existence of microcalcifications, masses, asymmetries, and distortions which are hidden in a noisy texture of breast tissue, glands, and fat. Along with the noisy data, they must analyze large amounts of mammograms yearly [1], with only about 0.5% containing cancerous structures [18]. Because of the large amount of negative mammograms, radiologist may become less acclimated to detecting subtle signs of breast cancer. Computer aided diagnosis (CAD) provides a second look in the mammogram screen process. The radiologist is prompted with regions of interest which can increase classification accuracy and screening efficiency. Because the anomalies exist within highly homogenous fatty tissue and glands, it is a non-trivial task to detect and locate them. Texture analysis in this field allows for a detection method that does not depend on a distinctively shaped growth.

As we shall show in the experimental section, measures based on the CK method allows us to classify and cluster nematodes and other datasets with great accuracy and speed, without the need (indeed, without the ability) to fine tune many parameters. We further show the generality of our ideas with a comprehensive set of experiments.

The rest of this paper is organized as follows. Section 2 contains a discussion of related and background work. In Section 3 we introduce our novel CK method and, the MPEG-1 video compression employing, CK-1 measure. In Section 4 we give details of the most obvious rival methods before we consider the most extensive set of experiments ever attempted for texture measures, in Section 5. In section 6, we provide a speed performance evaluation for the presented methods. Finally, in Section 7 we offer conclusions and a discussion of avenues for future research.

## 2. RELATED WORK / BACKGROUND

### 2.1 A Brief Review of Texture Measures

The measurement of texture similarity has a three-decade history and is still the subject of active research, see [33] and the references therein for an excellent overview. In essence, most methods reduce to some method to *extract* features combined with some measure to *compare* features.

These features can be *global scalars* such as energy, entropy, autocorrelation, standard deviation, etc., *global vectors* such as wavelet coefficients, Fourier coefficients, etc., or *local vectors/sets* such as SIFT descriptors, textons, etc.

The distance measures between the features are also highly variable, and include Euclidean distance,

Kullback distance, Dynamic Time (histogram) Warping, and the Earth Movers Distance [41]. Note that if the feature vectors/feature sets can be of different lengths, then we are forced to use an “elastic” distance measure that allows non-linear mappings for comparison of features. Note that such measures invariably have at least quadratic time complexity [41], often with high constant factors.

Beyond computer science led research efforts, we have noted that many real-world practitioners in biological domains simply extract many features, feed them into a neural network, and hope for the best [22][31][42]. Our informal survey suggests that this use of neural networks is often a last resort effort that comes at the end of frustrated attempts to deal with the huge combination of features/measures. As we shall later show, the CK-1 measure typically outperforms these efforts with a technique that is much simpler and orders of magnitude faster.

## 2.2 Kolmogorov Complexity Inspired Distance Measures

The CK method is based on recent pragmatic work which exploits the theoretical concepts of Kolmogorov complexity. Kolmogorov complexity is a measure of randomness of strings based on their information content. It was proposed by A.N. Kolmogorov in 1965 to quantify the randomness of strings and other discrete objects in an objective manner.

The Kolmogorov complexity  $K(x)$  of a string  $x$  is defined as the length of the shortest program capable of producing  $x$  on a universal computer — such as a Turing machine. Different programming languages will give rise to distinct values of  $K(x)$ , but one can prove that the differences are only up to a fixed additive constant. Intuitively,  $K(x)$  is the minimal quantity of information required to generate the string  $x$  by a program.

In order to define a distance based on the Kolmogorov complexity, the notion of conditional complexity is introduced. The conditional Kolmogorov complexity  $K(x|y)$  of  $x$  to  $y$  is defined as the length of the shortest program that computes  $x$  when  $y$  is given as an auxiliary input to the program. In [28], a distance is defined by comparing the conditional complexities  $K(x|y)$  and  $K(y|x)$  to  $K(xy)$ , the latter of which is the length of the shortest program that outputs  $y$  concatenated to  $x$ . More precisely, the authors define the distance  $d_k$  between two strings  $x$  and  $y$  as:

$$d_k(x, y) = \frac{K(x|y) + K(y|x)}{K(xy)} \quad (1)$$

The distance measure is completely parameter-free (it is independent of the computer language used) and has been shown to be optimal [29] in the sense that it subsumes other measures. Unfortunately, the

Kolmogorov complexity is incomputable for virtually all strings and thus must be approximated.

It is easy to see that universal compression algorithms give approximations to the Kolmogorov complexity. In fact,  $K(x)$  is the best compression that one could possibly achieve for the text string  $x$ . Given a data compression algorithm, we define  $C(x)$  as the size of the compressed  $x$  and  $C(x|y)$  as the compression size achieved by first training the compressor on  $y$ , and then compressing  $x$ . For example, if the compressor is based on a textual substitution method, one could build the dictionary on  $y$ , and then use that dictionary to compress  $x$ .

We can approximate the distance  $d_k$  by the following distance measure:

$$d_c(x, y) = \frac{C(x|y) + C(y|x)}{C(xy)} \quad (2)$$

The better the compression algorithm, the better the approximation of  $d_c$  is for  $d_k$ . In recent years this idea has been applied to domains as diverse as discovering the evolutionary histories of chain letters, spam classification, alignment-free comparison of biological sequences, protein structure classification [24], plagiarism detection [4], music genre classification, and a host of other problems [29].

Unfortunately, we cannot directly leverage on this body of work for two reasons. The first is that these ideas are only defined for *discrete* data, such as DNA strings or natural language. In these domains, a lossless compressor can really take advantage of repeated structure, which is exactly what we want to find to measure similarity. However, with the trivial exceptions such as cartoons/clip art, etc., most interesting images are *real-valued*. This difference is telling because lossless compression of discrete data is well defined and trivial to measure. In contrast, lossless compression of real-value images typically does reduce the sizes of the files greatly, but not in a way that finds repeated structure that is indicative of similarity.

The second reason we cannot directly use these ideas is more pragmatic. Calculating  $C(x|y)$  requires a detailed understanding of the compression algorithm  $C$ , and actually “hacking” into it. While such work would not be beyond a reasonable attempt, it is not within the scope of effort for us in conducting this research. It would limit the adoption of our ideas, especially among domain experts that are not computer scientists.

To solve these two problems we propose a modification of the  $d_c$  (and therefore  $d_k$ ) distance measure which treats a *lossy* compression algorithm as a complete black box, and which works for large, real-valued image data. In the Section 3 we expound these ideas.

## 2.3 Other Kolmogorov-Based Measures

To the best of our knowledge, this is the first work to consider compression-based distance measures for texture matching. A recent work considers a compression-based distance measure for *color* distributions in images [32], a paper by Li<sup>1</sup> and Zhu attempts image classification based on a kernel LZ78-based string kernel [37], Cilibrasi and Vitanyi create a compressor for clustering hand written text [8], and a recent work by Cerra and Datcu use a compression based measure for classifying satellite photographs [10].

However, beyond not explicitly considering texture, one thing all these works have in common is that they linearize the images into strings, and define distance measures based on strings. An obvious problem with converting a two-dimensional image into a one-dimensional string is that all spatial localization is lost. This may make no difference for color; however the very definition of texture is tied up with spatial patterns.

A recent paper proposes a compression based measure for similarity retrieval of ornamental letters in historical manuscripts (although compression-based, the authors do not make the connection to Kolmogorov inspired methods) [11]. The distance measure is based on the similarity of the run-length-encoding representations of the data. While the idea is interesting, the measure requires careful alignment of the two objects being compared and is only defined for binary images. Either restriction would prevent us using the measure on 90% of the datasets we consider in this work.

## 3. THE CK METHOD AND CK-1 MEASURE

In this section we give the high-level intuition behind the CK method of utilizing video compression for texture analysis and the CK-1 distance measure which utilizes MPEG-1 video encoding. We then give the concrete algorithmic details and conclude with explicit implementation aspects.

### 3.1 Intuition behind our Method

Recall that our basic goal, motivated by the successful use of compression-based distance measures in discrete-valued data mining domains [16][28][29], is to somehow exploit compression for measuring texture similarity in real-valued images. Whatever solution we come up with, we are very hesitant to deeply “hack” into image compression code. This reluctance here is

not mere sloth on our part, it is simply the case that difficult to implement ideas are rarely widely adopted. We feel that this is particularly true in this case, because much of our intended audience is biologists, nematologists, arachnologists, entomologists, etc. That is to say, people who may be comfortable using computer tools but are unlikely to have the time or the skills to write complex image compression code.

With this in mind we are motivated to use existing tools if possible. This leads us to consider measuring *image* similarity by exploiting *video* compression. Video is simply a three-dimensional array of images. Two dimensions, horizontal and vertical, serve as spatial image information directions of the moving pictures and the remaining dimension represents what is normally the time domain.

Virtually all video data contains significant amounts of spatial and temporal redundancy. Thus most video representations exploit these redundancies to reduce the file’s size. Similarities are encoded by merely registering differences within a frame (intra frame compression), and/or between frames (inter frame compression). Our idea then is to exploit video compression for measuring the similarity of two images, simply by creating a synthetic “video” which is comprised of the two images to be compared. If those two images are indeed similar, the inter frame compression step should be able to exploit that to produce a smaller file size, which we will interpret as significant similarity.

While there are dozens of video formats in existence, we choose MPEG-1 and refer to its use with the CK method as the CK-1 measure. We utilize MPEG-1 encoding because of its widespread availability and the fact that all implementations of it tend to be highly optimized. In the next section we will review the necessary details of MPEG-1 encoding.

### 3.2 MPEG-1 Encoding

Because the MPEG-1 specification allows variable application based implementation of spatial redundancy reduction and motion vector calculation for temporal redundancy reduction [12][26], we choose to utilize the MPEG-1 encoder provided by MathWorks in Matlab for its simplicity and availability. We use a consistent set of encoder parameters based on empirically verified intuitions. Empirical tests have illustrated that deviation from the following encoder parameters has either drastically reduced classification accuracy or has only shown negligible improvement for a small subset of the data sets.

For speed and consistency, a logarithmic search algorithm is utilized for the inter frame block matching process. Original images for intra-picture reference frames are used to bypass their encoding step. The

---

<sup>1</sup> This *Ming Li* [37] should not be confused with the *Ming Li* [28][29][30] who is a pioneer of Kolmogorov inspired distance measures.

resulting full quality reference frame also allows for more detailed texture matching by creating a precise “dictionary” of textures from the original image. Since we are only interested in the compression ratios of the images rather than their visual presentation, large quantization scales for reference(I) and predicted(P) frames are selected to prefer compressibility over image quality. This down samples the images and removes subtle differences between textures that may simply be attributed to noise. Since there are no bidirectional (B) frames in our usage, their quantization factor is ignored. The default Matlab search radius of ten pixels is maintained.

The bits used to specify block matched motion vectors have been limited to two. This modification is to allow for the possibility an exhaustive block match search and global references which may be too distant from the query block (would require more bits to reference than to store the original data), but has no affect on our reported results. The utility of global motion compensation is further discussed in section 7.

### 3.3 Video Creation

In our function, *mpegSize*, we use the MPEG-1 encoder to construct a video of two images. This function requires two images which are converted to grayscale for color invariance. Each image is then transformed into a Matlab movie frame. Then, an ordered Matlab movie is constructed with these two frames. This Matlab movie is subsequently passed to the MPEG-1 encoder. For speedup, we modify the encoder to bypass disk writes and simply return the resulting size of the MPEG-1 movie. The first image supplied to *mpegSize* is assigned as an **I** frame and the second becomes a **P** frame. Because the second image is compressed to references of the first, this function is not symmetric.

### 3.4 CK-1 Distance Measure

As hinted at in Section 3.1, in order to measure the distance between two images we analyze compression ratios. Our measure is accomplished with a simple equation:

$$d_{mpeg}(x, y) = \frac{C(x|y) + C(y|x)}{C(x|x) + C(y|y)} - 1 \quad (3)$$

As shown in Table 1, this is executed on two images *x* and *y* by just a single line of Matlab code:

```
function distance = CK1Distance(x, y)
distance = ( ( mpegSize(x, y) + mpegSize(y, x) ) / ...
( mpegSize(x, x) + mpegSize(y, y) ) ) - 1;
```

**Table 1: Our proposed distance measure**

Our CK-1 distance measure exhibits both positive definiteness and symmetry.

#### 3.4.1 Positive Definiteness

The CK-1 distance measure exhibits non-negativity. Given the consistency of our *mpegSize* function, the CK-1 distance of an image to itself will be zero. This property is important because many clustering algorithms rely on it to prove convergence properties.

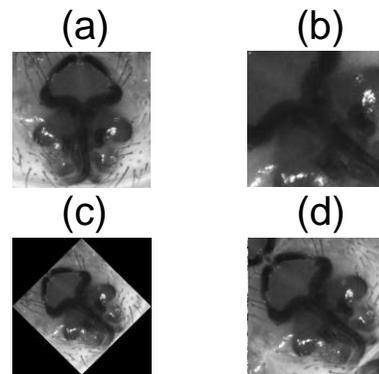
#### 3.4.2 Symmetry

As stated, our *mpegSize* function is not symmetric. To build a distance measure with symmetry, the bidirectional sum of the distances is taken in the numerator of (3) and the sum of the lower bounding sizes is in the denominator.

In addition, preprocessing techniques can be applied to the images to introduce several additional invariances to our approach. In our experiments we may utilize methods to achieve rotation, color, and illumination invariance.

#### 3.4.3 Rotation Invariance

For rotation invariance we fix one image and rotate the other to find the minimum CK-1 distance between them. When an image is rotated not at a 90°, 180°, or 270° angle, the image no longer fits into its original rectangular dimensions and a sampling method must be used. In our experiments we utilize three processes: no cropping, cropping to original image dimensions, and center cropping to a minimum bounding rectangle of valid pixels; black pixel padding or mirroring schemes are also used when rotations incur additional image pixels. Figure 3 demonstrates examples of these methods. Though different rotation methods provide better accuracies in different datasets, to avoid over fitting, we only report the accuracy provided by the center cropping method. For further simplification, we only consider ten rotations of the image in reported results; though our measure is fast enough to consider many more rotation degrees (cf. Appendix B).



**Figure 3 - Sampling and padding methods. (a) Original image, (b) center crop, (c) zero padded with larger dimensions, and (d) mirrored with original dimensions**

### 3.4.4 Color Invariance

We remove color information and analyze the textures based on their gray scale intensity values. For datasets where color information is useful, we could combine the CK-1 measure with color features [49].

### 3.4.5 Illumination Invariance

Illumination between images may vary between photographs of samples, with different cameras, locations, and photographers. To remove the inconsistencies due to lighting we normalize the intensity values of the images. For local illumination invariance due to shadows from edges and surface texture, we can normalize the intensity values across an entire image. We can then normalize between two images for inter-image illumination invariance. For simplicity, our results presented in this paper refrain from exploiting any accuracy improvements provided by these preprocessing techniques.

## 4. RIVAL METHODS

In this section we give concrete details of the most frequently used texture measures, as these will be the baseline to which we compare our ideas.

### 4.1 Filter Banks

The use of filter banks for feature extraction of textures has been motivated by their ability to be tuned to many diverse applications [22][35][42]. Their utility has allowed for a wide spread use in computer vision applications with many high-quality results. While there are many possible filter banks, the Gabor filter is by far the most commonly used. An overview of Gabor filters can be found in [3][34][50][39]. To generate our filters, a mother wavelet and generation function as presented in [50] is utilized. Filters of six orientations and four scales are generated, resulting in a filter bank of size  $N = 24$  filters. High and low frequency parameters of the filters were set to the specifications found in [50].

Images are convolved with each filter. The standard deviation and mean of each response is then aggregated into a single 48 length vector. The distance between image descriptors can then be found from their Euclidean distance.

### 4.2 Textons

In order to fairly compare our method, we take the extra step of extending the previously described filter bank approach by classifying with a dictionary of representative filter responses, *textons*. Textons have been shown to be a great improvement over basic filter bank techniques [27][47]. Following the texton dictionary creation of [47], we represent each pixel of an image by a response vector of its corresponding outputs from each of the 24 filters. Response vectors

from all images within a single class are then clustered into ten groups using kmeans clustering, provided with Matlab, and the centroids of these clusters from each class are added to the texton dictionary. An image can then be represented by its histogram of response vectors binned to the nearest texton in the texton dictionary. The distance between two texton histograms is then found using the chi-squared distance.

## 5. EXPERIMENTAL EVALUATION

We begin by stating our experimental philosophy. To ensure that our experiments are not just reproducible, but *easily* reproducible, we have built a website which contains all data and code, together with the raw spreadsheets for the results [6]. In addition this website contains additional experiments that are omitted here for brevity.

### 5.1 Sanity Check

We begin with a simple experiment on a domain where human intuition can directly judge the effectiveness of the CK-1 measure.

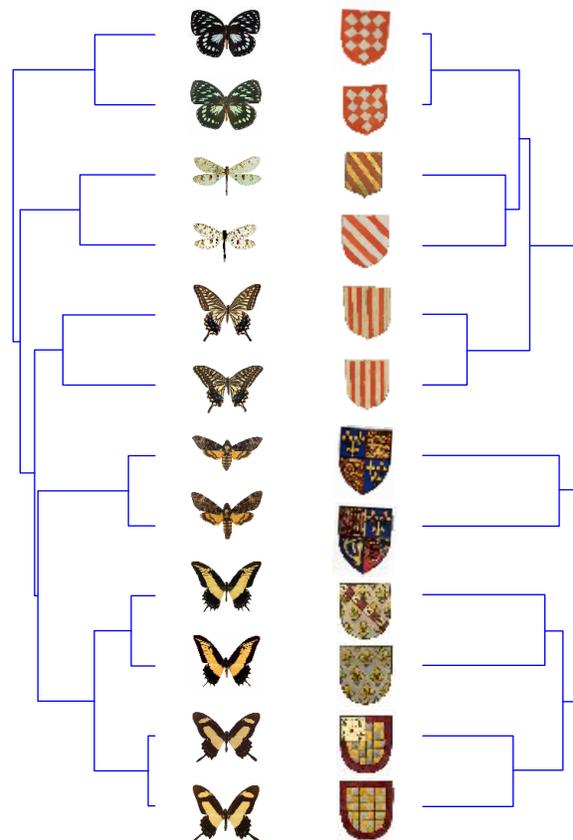


Figure 4: The *Insect* dataset and *Heraldic shields* datasets clustered with the CK-1 distance measure (average linkage clustering). While the images are shown in color for clarity, our distance measure had only access to the grayscale version of the images

We clustered two sets of images, both of which have previously been used to test the utility of color and shape distance measures [49]. The two datasets are: *Heraldic shields* extracted from historical manuscripts from the 14<sup>th</sup> to 16<sup>th</sup> century, and *Insects* extracted from various amateur entomologists websites (used with permission). In both cases we selected 12 images which could be objectively or subjectively sorted into six pairs, Figure 4 shows the results.

Compared to previous work, the results are unexpectedly good. In past work we had clustered (supersets) of these datasets based on color (shields) and color/shape (insects), but ignored the texture because we assumed it would not be very useful [49]. To our surprise, right “out of the box” the compression-based measure works much better than our carefully tuned color/shape measure [49].

## 5.2 Classification Experiments

In order to demonstrate the generality of our methods we have assembled a large and diverse collection of datasets. The descriptions below are necessarily brief; for more details we refer the interested reader to Appendix A, the supporting webpage [6], or the originating papers. Note that in every case we make these datasets publicly available (with the copyright remaining with the original creators where appropriate). The smaller datasets can be downloaded from [6]; the entire dataset can be obtained on two free DVDs by emailing the second author. In Figure 5 we show examples from each dataset.

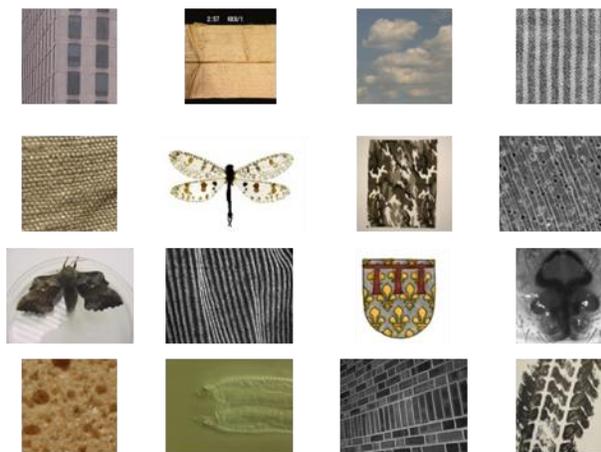


Figure 5: Samples of the datasets considered. A detailed key is omitted here for brevity, see [6]

**Arachnology (Spiders):** This dataset [42] consists of images of the Australasian ground spiders of the family *Trochanteriidae*. This is a diverse family with high variance in inter- and intra-specific variation and sparse representation of the classes.

**Moths (Macrolepidoptera):** This collection [35] consisting of the images of live moth individuals, each moth belonging to one of 35 different species found in the British Isles. We consider three variants of this dataset: the original data, in which the moth occupies about 10% of the image area; center cropped, where an approximate bounding box was placed around the image; and a cleaned version, where the background was deleted with a semi-automatic technique.

**Tire Treads:** This dataset consists of a collection of tire imprints left on paper. Three well worn tires had paint applied to their treads and were rolled over paper. The tires are painted and rolled 16 times, each in varying directions and with different painted sections of the tire.

**Nematodes:** As noted in the introduction, nematodes are a diverse phylum of “wormlike” animals, with great commercial and medical importance. The department of nematology at UCR, one of the leading institutions of in nematode research, has recently tasked us with creating a distance measure to help them sort through the largest archive of high-quality nematode images in the world [14].

**Brodatz Textures:** This dataset consists of a diverse set of images of man-made and natural textures (grass, straw, cloth, etc.), digitalized from images from a reference photographic album for artists and designers. Our version was obtained mostly from a publicly available online image database [40]. This set was missing slate 14, which we added directly from an original copy of the text held at our campus library [5].

**CAIRO Wood Set:** This dataset consists of 100 images of ten species of tropical wood provided by the Center for Artificial Intelligence and Robotics [7]. Each species is represented by ten photographs taken at a microscopic level. The images are also evenly split into two families of wood, *Leguminosae* and *Dipterocarpaceae*. The dataset is classified in two approaches: a two-class problem across family designations and a ten-class problem across species classifications.

**Camouflage:** This dataset consists of 70 images of nine varieties of modern US military camouflage. The images were created by photographing military t-shirts and fabrics at random orientations.

**VVT Wood Set:** This dataset consists of wood images originally for color based inspection and grading for industrial usage [43]. Images of wood are tessellated and classified into 40 types of wood defect (dry knot, small knot, bark pocket, core stripe, etc.). The annotated data is parsed and each tessellated region is cropped and given a class label of either sound or

defective. We use a subset consisting of 100 images from each class for classification tests.

**UIUCTEX:** The UIUCTEX data set [25] is composed of images of common textures such as glass, bark, and water. They are taken at varying orientations, illuminations, and subset locations on the sample texture.

**VisTex:** The MIT Vision Texture data set [36], unlike many other texture datasets, does not hold rigid rules for orientation or lighting. Rather, it provides images from real world conditions such as flowers within a field or the water texture from an inland position.

**KTH-TIPS:** The KTH-TIPS [19] texture data set exists as an extension of the CUREt data set [9] by adding variances in scale and by photographing from multiple samples in a single class.

Data Set	CK-1 (%)	RI CK-1 (%)	Gabor Filters (%)	Texton (%)
Spider Subset	<b>96.3</b>	-	59.6	89.6
Full Spider Set	<b>93.1</b>	-	39.1	74.1
Tire Tracks	79.2	91.7	87.5	<b>93.8</b>
Nematodes	<b>56.0</b>	-	38.0	52.0
CAIRO Wood (F)	83.0	94.0	<b>95.0</b>	<b>95.0</b>
CAIRO Wood (S)	77.0	90.0	93.0	<b>94.0</b>
VTT Wood	81.5	<b>92.0</b>	88.0	89.5
Original Moths	<b>49.1</b>	-	18.3	42.6
Cropped Moths	<b>63.4</b>	-	27.5	48.8
Cleaned Moths	<b>71.0</b>	-	24.0	58.2
Brodatz	<b>52.1</b>	44.8	37.0	52.0
KTH-TIPS	<b>73.7</b>	63.3	58.3	54.8
Camouflage	87.5	-	85.0	<b>92.5</b>
UIUCTex	51.0	43.6	45.3	<b>55.8</b>
VisTex	32.9	26.3	36.5	<b>47.9</b>

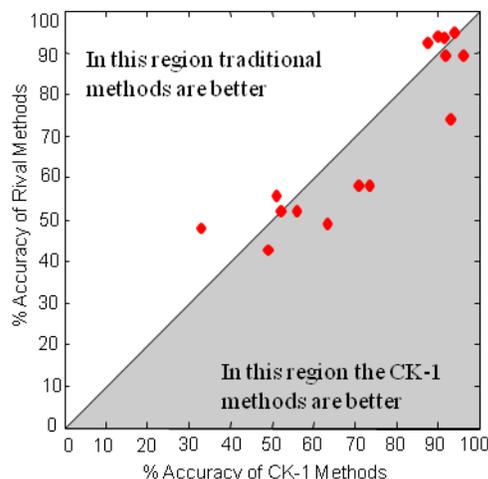
**Table 2: Accuracy of the one-nearest-neighbor classifier using the four measures under consideration. Note that results may be biased towards the texton approach. Also, for registered data sets we did not consider the rotation-invariant CK-1 measure**

We test all algorithms by doing leaving-one-out classification with the one-nearest neighbor algorithm. For the relatively slow Texton approach (cf. Figure 8), these experiments would take years if we had to relearn the Texton dictionary on each fold. We therefore allowed the Texton method to “cheat” by learning the dictionary on the entire dataset. As such, the results for the Texton method may be optimistic.

Table 2 presents the best experimental results for these data sets with the CK-1 measure, the rotation invariant CK-1 measure, the Gabor filter bank method, and the texton method.

Because the sheer number of results makes it difficult to judge the relative performance of the distance

measures, we produced a figure to help visualize the results. For each dataset, we created a variable  $X = \max(\text{CK-1}, \text{RI CK-1})$ , and a variable  $Y = \max(\text{Gabor Filters}, \text{Textons})$ ; we used these variables to plot a point for each dataset in Figure 6.



**Figure 6: A visual summary of the relative strength effectiveness of our proposed distance measure**

Here we can see at a glance that the CK methods are extremely effective (Recall that classifications are biased towards the texton measure due to its learning on the entire dataset).

### 5.3 An Application to Web Mining

We conclude our experiments with a simple example of a web mining application that can benefit from a robust texture measure. Our experiment is somewhat contrived, but demonstrates the robustness of the CK-1 distance to general unseen and unstructured data.



**Figure 7: A web query for *munda did* produce some images of the moth, *Orthosia munda*, we expected (left), but it also returned images of the Munda tribes of India (top left), a map of Munda Island (top right), an unrelated insect *Cycloneda munda* (bottom left) and a military photo taken at Munda Island (bottom right)**

While gathering datasets for the classification experiments in the previous section, we noted we had a folder of moth images simply labeled *munda* (we know now the Genus name is *Orthosia*). Suppose we wished to retrieve more images of these moths from web, we

can simply issue a Google image search. We did this on October 4<sup>th</sup>, 2009 and found that of the twenty-one images returned on the first page, none showed the correct moth. An image of the moth could not be found until the second page and the next image of the moth did not appear until the third page. As shown in Figure 7, the false positives include images of Munda Island and an unrelated insect that has the same specific name. For simplicity, let us consider the first four pages, which consist of 84 images, as the entire universe of images. Considering only these pages, there is a precision and recall of zero on the first page. There is an obvious way we could increase the precision of the query in the first page of results. Since we have some images of the moth we are interested in we could issue the text query as before, then reorder the query results based on their distance to a representative of our training data. This training representative is the training image with the lowest mean CK-1 distance to all other training images. We then score each query image based on their CK-1 distances to this training representative. This reordering brought about a recall of 1.0 and a precision of 0.19 on the first page.

## 6. RUNTIME PERFORMANCE

The speed of our CK based method can be attributed to the simplicity of the underlying MPEG-1 compression algorithm. Since the reference image is not down sampled, there is no time required for its spatial redundancy reduction. The most time costly process, interframe block matching, is a logarithmic search process. Also, each block in the query image need only be compared to its corresponding neighborhood in the reference image. This greatly limits the running time needed to block match an entire image to  $O(n \log n)$ . Because the search can *early abandon* depending on the quality of a found match, this worst case runtime is usually avoided in empirical tests in favor of a fast average case runtime. Furthermore, since most uses of MPEG involve large movies in the commercially important entertainment industry, the MPEG compression algorithms are extraordinarily well optimized.

In contrast, Gabor filters must convolve  $N$  filters for each image. The time performance of this operation must then also consider the dimension  $D$  of the square filters, where  $D \gg N$ . The size of  $D$  depends on the scale and frequency parameters used in the filter generation and, in some cases, can be larger than the image itself. Just the Gabor descriptor extraction is therefore an  $O(n^2)$  operation.

Textons add onto the running time of the original Gabor filters approach by requiring clustering within each class. Its runtime is bounded by  $O(n^2) + n \times (\text{images per class}) \times (\text{number of classes})$ , where each

element to be clustered is of  $N$  dimensions. Texton calculation speed performance is therefore heavily dependent on its application. Large numbers of classes, large images, and large collections of images can greatly increase the execution time.

As a concrete example: the distance between two images from the VisTex dataset, grass and brick, are compared with each of the three methods. The distances of ten scales of these images are computed and the average execution times over several iterations are plotted in Figure 8.

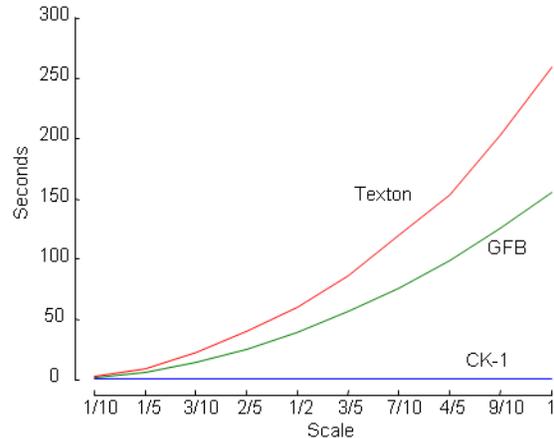


Figure 8: Time comparison of CK-1, Gabor Filter Banks (GFB), and Texton approaches

As we can see, the time taken for the CK-1 measure is negligible relative to the other measures.

## 7. CONCLUSION AND FUTURE WORK

In general the results in the previous section speak for themselves. For the most part, we have avoided comparisons to published results that consider the same datasets since different experimental conditions make direct comparisons difficult. However in some cases tentative comparisons can be instructive.

In the Spider Subset problem we got an accuracy of 96.3%, the original authors obtained accuracy in “the range of 90–96%” [42]. Note that this range of accuracy was obtained at the end of a four-year project devoted to just this problem, and their algorithm required occasional human intervention, “it was important to review the log files of this process to pick out any potentially contaminating images and remove them from the training sets” [42].

Of the variants of the Moth dataset, we obtained a best accuracy of 71.0%. Using two variants of the Nearest Neighbor algorithm (as we did), the original authors obtained 65.7 and 71.6% respectively [35]. However it is important to note that we used *only* texture features, whereas the original work had access to both color and texture features. It is clear that color is very useful in

discriminating at least some of the classes. For example *Ourapteryx sambucaria* is yellow, whereas *Campaea margaritata* gets its common name, the Light Emerald moth, from its distinctive green hue, and *Cabera pusaria* is aptly known as the Common White Wave.

It is important to note that in spite of the generally excellent performance of the CK-1 measure in diverse domains, we are not claiming it is the best measure possible for all problems. For specialized application areas, better measures, which incorporate domain specific constraints and features, *may* do better. However for exploratory data mining, our CK-1 measure, built on our CK method, offers a powerful yet simple baseline measure.

### 7.1 Future Work

In this work we have not focused on the speed or indexability of the CK-1 measure. One reason for this is that we wanted to forcefully demonstrate its *utility* first. In addition, we feel that optimizing speed may be irrelevant in many domains. Theo Pavlidis, one of the founders of CBIR recently remarked, “*In a medical application it may take well over an hour to produce an image, so waiting another hour to find matches in a database is not particularly onerous*” [38]. Such remarks apply to many of our domains; the moth dataset took almost a year to collect and the nematode dataset took four years to collect [14][35].

Nevertheless, as we have shown in Figure 8, the CK-1 measure is orders of magnitudes faster than some obvious rivals.

Still, there may be data mining applications for which we need to further improve efficiency. For example, within the next two years we expect to have terabytes of nematode images [14].

There are several possibilities we plan to pursue. One possibility is to modify the *measure* so that it becomes a *metric*. This would allow us to avail of a wealth of techniques that exploit the triangular inequality to index data.

Further improvements in speed may come from exploiting several known ideas in image/video processing. For example multi-resolution analysis for scale invariance could improve our method’s performances in many domains. More advanced compression algorithms could be explored to be used with the CK method for possible performance increases in speed and accuracy. Modifying the block matching search algorithm to allow for global motion vectors could allow for higher accuracies or faster search procedures and batch processing of multiple images. Possible options include the creation of a block matching algorithm specifically for the application of texture analysis, or to explore the global compensation

techniques implemented in newer compression methods such as MPEG-4 and H.264 [48].

### 8. ACKNOWLEDGMENTS

This work was funded by NSF 0808770. We would like to thank the many donors of data, especially of Anna Watson and Michael Mayo (moths), Melissa Yoder and Paul De Ley (nematodes), Kimberly Russell (spiders), and Marzuki Khalid (CAIRO Wood).

### 9. REFERENCES

- [1] American Cancer Society, *Breast Cancer Facts & Figures 2007-2008*, American Cancer Society, Inc.
- [2] J. Bethony, S. Brooker, M. Albonico, S. Geiger, A. Loukas, D. Diement, P. J. Hotez, *Soil-transmitted helminth infections: ascariasis, trichuriasis, and hookworm*. The Lancet, 2006.
- [3] F. Bianconi, A. Fernandez, *Evaluation of the effects of Gabor filter parameters on texture classification*. Pattern Recognition 40(12), 3325–3335 (2007).
- [4] A. Bratko, G. Cormack, B. Filipic, T. Lynam, B. Zupan, *Spam Filtering Using Statistical Data Compression Models*, Journal of Machine Learning Research 7, Dec. 2006.
- [5] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, New York: Dover, 1966.
- [6] B. Campana, Website for this paper <http://www.cs.ucr.edu/~bcampana/texture.html>.
- [7] Center for Artificial Intelligence and Robotics, <http://www.cairo-aisb.com/>.
- [8] R. Cilibrasi, P. Vitányi, *Clustering by Compression*, IEEE Transactions on Information Theory 51, 1523-1545, 2005.
- [9] K. J. Dana, B. van Ginneken, S. K. Nayar, J. J. Koenderink, *Reflectance and texture of real-world surfaces*, ACM Trans. Graph. 18, 1, 1999.
- [10] D. Cerra, M. Datcu, *A Model Conditioned Data Compression Based Similarity Measure*, DCC, 2008.
- [11] M. Delalandre, J. Ogier, J. Lladós, *A fast cbir system of old ornamental letter*. In Workshop on Graphics Recognition (GREC), vol. 5046 of Lecture Note in Computer Science (LNCS), pages 135–144, 2008.
- [12] Coding of moving pictures and associated audio. *Committee Draft of Standard ISO 11172: ISO/MPEG 90/176*, Dec. 1990.
- [13] P.R. Cohen, D. Jensen, *Overfitting explained*, In Prelim. Papers Sixth Intl. Workshop on Artificial Intelligence and Statistics, pages 115–122, January 1997.

- [14] P. De Ley, Assistant Professor and Assistant Nematologist at UCR, Personal communication, Feb 2009.
- [15] S.W. Duffy, L. Tabar, H. H. Chen, et al, *The impact of organized mammography service screening on breast carcinoma mortality in seven Swedish counties*, *Cancer*, 95(3):458-469, Aug 1 2002.
- [16] E. J. Keogh, S. Lonardi, C. A. Ratanamahatana, L. Wei, S. Lee, J. Handley, *Compression-based data mining of sequential data*, *Data Min. Knowl. Discov.* 14(1): 99-129, 2007.
- [17] M. Garcia, A. Jemal, E. M. Ward, M. M. Center, Y. Hao, R. L. Siegel, M.J. Thun, *Global Cancer Facts & Figures 2007*, Atlanta, GA: American Cancer Society, 2007.
- [18] M. Giger, *Current Issues in CAD Mammography. Digital Mammography*, Proc. 3rd Int. Workshop of Digital Mammography, 1996.
- [19] E. Hayman, B. Caputo, M. Fritz, J. O. Eklundh, *On the significance of real-world conditions for material classification*, In: Proc. European Conf. on Computer Vision, No. 3, Springer-Verlag, pp. 253-266, 2004.
- [20] L. L. Humphrey, M. Helfand, B.K. Chan, S.H. Woolf, *Breast cancer screening: a summary of the evidence for the U.S. Preventive Services Task Force*, *Ann Intern Med.*, 137 (5 Part 1):347-360, Sep 3 2002.
- [21] A. Jemal, R. Siegel, E. Ward, et al, *Cancer Statistics*, *CA Cancer J Clin.*, 56:106-130, 2006.
- [22] I. Kavdir, *Discrimination of sunflower, weed and soil by artificial neural networks*, *Computers and Electronics in Agriculture* 44, 153–160, 2004.
- [23] M. Khalid, *Design of an Intelligent Wood Species Recognition System*, Technical Report of Center for Artificial Intelligence and Robotics (CAIRO), Malaysia, 2008.
- [24] N. Krasnogor, D. A. Pelta, *Measuring the similarity of protein structures by means of the universal similarity metric*, *Bioinformatics*, 20, : 1015–1021, 2004.
- [25] S. Lazebnik, C. Schmid, J. Ponce, *A Sparse Texture Representation Using Local Affine Regions*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265-1278, August 2005.
- [26] D. Le Gall, *Mpeg: a video compression standard for multimedia application*, *Commun. ACM*, vol. 34, no. 4, pp. 46-58, April 1991.
- [27] T. Leung, J. Malik, *Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons*, *Int. J. Comput. Vision* 43, 1, 29-44. Jun. 2001.
- [28] M. Li, J. H. Badger, X. Chen, S. Kwong, P. Kearney, H. Zhang, *An information-based sequence distance and its application to whole mitochondrial genome phylogeny*, *Bioinformatics* 17: 149-154, 2001.
- [29] M. Li, X. Chen, X. Li, B. Ma, P. Vitanyi, *The similarity metric*, *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Pages: 863 – 872, 2003.
- [30] M. Li, P. Vitanyi, *An Introduction to Kolmogorov Complexity and Its Applications*. Second Edition, Springer Verlag, 1997.
- [31] P. Li, J. R. Flenley, *Pollen texture identification using neural networks*, *Grana*, 38:59-64, 1999.
- [32] A. Macedonas, D. Besiris, G. Economou, S. Fotopoulos, *Dictionary based color image retrieval*, *Journal of Visual Communication and Image Representation*, v.19 n.7, p.464-470, October, 2008.
- [33] M. Mirmehdi, X. Xie, J. Suri, (eds.), *Handbook of Texture Analysis*, Imperial College Press., December 2008.
- [34] B. S. Manjunath, 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.
- [35] M. Mayo, A. Watson, *Automatic species identification of live moths*. In Ellis et. al, editor, *Proc. of the 26th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, 195-202, 2006.
- [36] MIT Vision and Modeling Group, <http://vismod.media.mit.edu/vismod/>.
- [37] M. Li, Y. Zhu, *Image Classification Via LZ78 Based String Kernel: A Comparative Study*, *PAKDD*, 704-712, 2006.
- [38] T. Pavlidis, *Limitations of CBIR* (Keynote Talk). *International Conference on Pattern Recognition*, 8-11, Tampa, Florida, Dec. 2006.
- [39] R. Porter, N. Canagarajah, *Robust Rotation-Invariant Texture Classification: Wavelet, Gabor Filter and GMRF Based Schemes*, *IEE Proc. - Vision, Image and Signal Processing*, 144:180-188, 1997.
- [40] T. Randen, *Brodatz Textures Image Database*, <http://www.ux.uis.no/~tranden/brodatz.html>.
- [41] Y. Rubner, C. Tomasi, L. Guibas, *The Earth Mover's Distance as a Metric for Image Retrieval*, *International Journal of Computer Vision*, v.40 n.2, p.99-121, Nov. 2000.

- [42] K. Russell, H. Do, J. Huff, N. Platnick, *Introducing SPIDA-web: wavelets, neural networks and Internet accessibility in an image-based automated identification system*, In N. MacLeod (ed), *Automated Object Identification in Systematics: Theory, Approaches, and Applications*. Springer Verlag, 2007.
- [43] O. Silven, M. Niskanen, H. Kauppinen, *Wood inspection with non-supervised clustering*, COST action E10 Workshop - Wood properties for industrial use, 18-22, Espoo, Finland, June 2000.
- [44] J. Suckling, *The Mammographic Image Analysis Society Digital Mammogram Database*, Excerpta Medica, International Congress Series 1069, pp375-378, 1994.
- [45] L. Tabar, M. F. Yen, B. Vitak, H. H. Chen, R. A. Smith, S. W. Duffy, *Mammography service screening and mortality in breast cancer patients: 20-year follow-up before and after introduction of screening*, *Lancet*, 1405-1410, Apr 26 2003.
- [46] J. Tou, P. Lau, Y. Tay. *Computer Vision-based Wood Recognition System*, Proceedings of International Workshop on Advanced Image Technology (IWAIT 2007), pp. 197-202, 2007.
- [47] M. Varma, A. Zisserman, *A Statistical Approach to Texture Classification from Single Images*, *Int. J. Comput. Vision* 62, 1-2, 61-81, Apr. 2005.
- [48] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, *Overview of the h.264/avc video coding standard*. *Circuits and Systems for Video Technology*, *IEEE Trans* 13(7):560–576, 2003.
- [49] X. Wang, L. Ye, E. J. Keogh, C. R. Shelton, *Annotating historical archives of images*, *JCDL*, 341-350, 2008.
- [50] P. Wu, B. S. Manjunath, S. Newsam, H. D. Shin, *A texture descriptor for browsing and similarity retrieval*, *Signal Processing: Image Communication*, Volume 16, Issues 1-2, Pages 33-43, September 2000.

## Appendix A: Dataset Details

In Table 3 we numerically summarize the datasets. *Image quality* is a subjective measure of how “clean” the images are, for example do they have occlusions on the subject or camera shake.

Data Set	Number of images	Number of classes	Image Size	Image Quality
Spider Subset	27	3	256x256	High
Full Spider Set	955	14	256x256	High
Tire Tracks	48	3	256x256	High
Nematodes	50	5	1440x1080	High
CAIRO Wood (F)	100	2	768x576	High
CAIRO Wood (S)	100	10	768x576	High
VTT Wood	200	2	~61x61	Medium
Original Moths	774	35	1280x960	Medium
Cropped Moths	774	35	800x800	Medium
Cleaned Moths	774	35	~500x800	High
Brodatz	1,792	112	128x128	High
KTH-TIPS	810	10	200x200	High
Camouflage	80	9	256x256	High
UIUCTex	1000	25	640x480	High
VisTex	334	19	512x512	High

Table 3: Dataset details

## Appendix B: Effects of Rotation

As noted in the main text we achieve rotation invariance by holding one image fixed and rotating the other. Since our measure is so fast we can quickly do this 360 times (once per degree) if necessary, however as hinted at in Figure 9, a coarser (and therefore faster search) is possible.

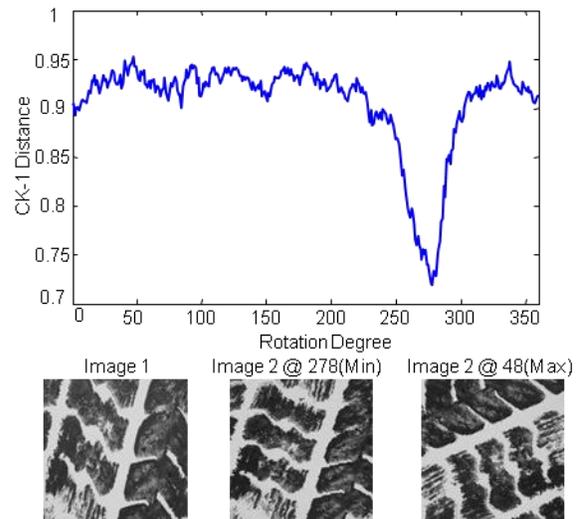


Figure 9: (Top) Measured CK-1 distance from image 1 to rotations of image 2. (Bottom) Center cropped images of image 1 and optimal and poorest rotations of image 2