The Randlet Transform

Applications to Universal Perceptual Hashing and Image Identification

Michael Malkin
mikeym@stanford.edu
Gates Building, Room 464
Stanford, CA 94305

Ramarathnam Venkatesan
venkie@microsoft.com
One Microsoft Way
Redmond, WA, 98052

Abstract

We introduce a new transform called the *randlet transform* and explore applications to universal perceptual image hashing (key-based randomized digests) and image identification. Our transform yields robust signal representations that are nearly invariant to several perceptually insignificant transformations (attacks, intentional or otherwise). Our signal representation is hard to guess without the secret key used in its derivation and is motivated by applications such as image identification and watermarking where attack resistance is important. It is also interesting in itself as a transform and in regular signal processing tasks such as scene detection, motion compensation algorithms, compression, etc.

We consider the performance of the randlet transform and the wavelet transform against attacks based on rotation, cropping, scaling, additive noise, and JPEG compression. The randlet transform achieves superior performance to the wavelet transform in the task of image identification. For example, a randlet transform with a false positive rate of 2% can detect a 10-degree rotation with a false negative rate of 3.2%, while the best wavelet transform with the same false positive rate has a false negative rate of 38.2%.

1 Introduction

The randlet transform is a new transform composed of randomly-chosen basis functions. This randomization gives the randlet transform distinct advantages which are important from signal processing and security points of view.

There are two types of randomization, explicit and implicit. With implicit randomization, input signals are assumed to be stochastic, and probabilistic claims are made about the output based on these assumptions. Signal processing applications generally use implicit randomization. Explicit randomization, like that used in the randlet transform, is randomization that is inherent to an algorithm and is independent of the input. Explicit randomization allows algorithms to avoid worst case performance and eliminates the need to make statistical assumptions about the input.

Because randlet transform basis functions are chosen at random from a large set of basis functions, the worst case performance of the transform occurs with extremely low probability. This is similar to the universal hash functions used extensively in the theory of computation and cryptography [2]. A universal hash function is chosen at random from a large family of hash functions. They are constructed in a way so that for any two distinct inputs (without any stochastic assumptions on the input) the probability (over the choice of hash functions) that the hash value collides is close to optimal. The main idea behind universal hash functions is that one may dispense with statistical assumptions about the input and for most choices of keys expect good performance. This is crucial since in the case of an intelligent, malicious attacker it is not always possible to make statistical assumptions about the input.

The randomness of the randlet transform also makes the transform coefficients independent (without the knowledge of the key) in a formal sense and allows analysis of the entropy of transform coefficients without making assumptions about the statistical nature of the input. Furthermore, there are benefits in security applications such as hashing and watermarking because an attacker does not know which basis functions are used in the transform, making attacks much more difficult.

It is natural to wonder why a new transform is necessary to introduce randomness. For example, a DCT can be multiplied by a random matrix to randomize the output. For many applications, though, randomization is not sufficient. In the case of image identification, the transform also must be *perceptual* to achieve good performance. A perceptual transform is one where two images that are perceptually similar will have similar transforms. The randlet transform is both randomized and perceptual, whereas the example of the randomized DCT is randomized but not perceptual.

Section 2 introduces the randlet transform. It is a family $\mathcal{F} = \{f_K\}$ of transforms indexed by a key K. Each transform uses a set of basis functions that are chosen pseudo-randomly based on the key, which in some applications can be held secret. The key specifies which member of the family will be used for a particular instance of the transform.

Basis functions in the randlet transform are called *randlets*. Randlets are generally formed by taking a small number of localized functions known as *mother randlets*, and scaling, translating, and rotating them. The result is a large number of possible randlets. Each randlet transform uses a small subset of all possible randlets as its basis functions. While they offer some multi-resolution properties, they do not have the fine structure (e.g., the nested vector spaces) as in standard multi-resolution analysis.

Section 3 discusses the performance of the randlet transform. The focus of this section is upon the use of the randlet transform in image hashing and image identification. We write that an image identification system $H(\cdot)$ takes as input an image I and outputs a vector v which represents the image: $v = H(I)$. Let $\mathcal{A}$ be a family of attacks against which H is meant to operate, and let $A \in \mathcal{A}$ be a specific attack. H has the property that when I is perturbed by an attack, v is not changed much, so $\|H(I) - H(A(I))\| \leq \epsilon$ with high probability. Also, if two distinct images are compared, then their vectors should be distinct as well, so $\|H(I) - H(I')\| > \epsilon$ with high probability. We define the following two terms:

Probability of false positive: $P_\epsilon^+ = \Pr[\,\|H(I) - H(I')\| \leq \epsilon]$
Probability of false negative: $P_\epsilon^- = \Pr[\,\|H(I) - H(A(I))\| > \epsilon]$

The goal of an image identification system is to achieve low P_ϵ^+ and P_ϵ^-, but lowering one will generally cause the other to rise. As epsilon is varied, P_ϵ^+ and P_ϵ^- describe an ROC curve[1]. The ROC curve itself is defined by the attack and the transform. An application will choose a point along the ROC curve according to its own particular needs. In general, it is more important to have a low P_ϵ^+ because in most systems there are many more distinct images than there are attacked images. Additionally, an adversary who is

[1]Generally, an ROC curve is defined as P_ϵ^+ versus $1 - P_\epsilon^-$. However, for our purposes it is more useful to have ROC curves of P_ϵ^+ versus P_ϵ^-.

attacking images may be deterred by a detection probability of even 50%.

When constructing an image hash, the vector v can be used by itself as the hash of an image, or it can be used as input to an error-correcting decoder. This will shrink the hash value and provide thresholding behavior, where a difference less than ϵ causes no change in the hash value, while a difference greater than ϵ results in a completely different and unrelated hash value.

Systems already exist that manipulate transform coefficients to perform image identification and image hashing (see section 1.1), but one of the motivations for the randlet transform was to generate a transform that was itself resistant to attacks. Additionally, the randlet transform can be used as a component in many of these other systems. This paper is concerned with the performance of transforms in identifying images, and so compares the randlet transform to the wavelet transform and not to these other, higher-level systems. We note, however, that the randlet transform, taken alone, achieves very good performance in relation to other systems as well.

In this paper our transforms are discrete and the focus is on algorithmic details and experimental results. Formal analysis and application to watermarking and compression will appear elsewhere[6]. We consider attacks based on rotation, cropping, scaling, additive noise, and JPEG compression. The randlet transform produces ROC curves superior to those of the DWT against most attacks, and especially against rotation attacks.

1.1 Previous Work

Matching pursuits [7] uses residuals and an overcomplete basis in a transform. Ventura et al. [5] use matching pursuits with two-dimensional Gaussian and Mexican hat basis functions. Curvelets [1] are another related transform.

Mihcak and Venkatesan [8] and Venkatesan et al. [10] use randomized methods for image hashing and identification, but they achieve their results by manipulating transform coefficients. Fridrich uses randomness in image hashing [3, 4], but the resulting transforms are not capable of inversion, and are susceptible to certain attacks, like rotation. Furthermore, these transforms are not localized and so can not take advantage of localized features of documents. Johnson et al. [9] attempt to extract image features such as edges and corners to identify images.

2 The Randlet Transform

The idea behind the randlet transform is to randomly generate a set of basis functions and project them onto an image. Without knowing the basis functions it is hard to predict what the projections will be.

Our transform can be inverted if sufficient number of coefficients are used. For identification or hashing purposes, 100 3-bit coefficients are generally sufficient to survive the attacks we tested. If reconstruction will be performed, or if it is important that the projections are orthogonal to each other, each projection is subtracted from the image before the next projection is made. Successive projections provide a closer approximation to the original image. The reconstruction is then simply the sum of each randlet multiplied by its transform coefficient. In the case of image hashing, no reconstruction will be performed, so the transform is simply the projection of each basis function onto the image.

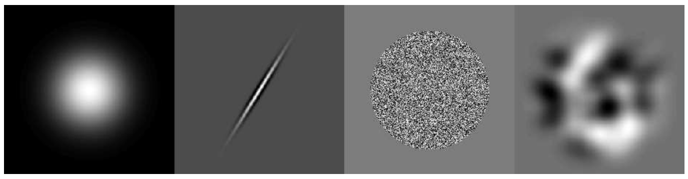


Figure 1: These are examples of randlets. From left to right they are: a 200×200 Gaussian randlet, a 200×14.14 Mexican Hat randlet, a 200×200 random randlet, and a 200×200 smooth random randlet.

2.1 What are Randlets?

Some example randlets are:

Gaussian Randlet: Translations and scalings of: $C \cdot e^{x^2+y^2}$.

Mexican Hat Randlet: Translations, scalings, and rotations of: $C \cdot y^2 e^{\sigma_x x^2 + \sigma_y y^2}$.

Random Randlet: Translations and scalings of an elliptical region where each pixel value is chosen uniformly from [0, 1], and the result is normalized. This randlet is not rotated, but many versions with different random pixels are generated.

Smooth Random Randlet: A random randlet that has been low-pass filtered.

Curvelet Randlet: Curvelets [1] may be used as randlets.

Wavelet Randlet: Translations, scalings, and rotations of: $C \cdot w(y) e^{\sigma_x x^2}$,where $w(y)$ is a wavelet.

Figure 1 shows some example randlets.

Mother randlets are two-dimensional functions with compact support or fast decay. In order to be perceptual, mother randlets should not contain high frequencies. Those that do, like random randlets, are not perceptual and give very poor performance.

Generally, a few functions are chosen as *mother randlets*, and these functions are randomly scaled, rotated, and translated to form a randlet transform. There are exceptions, as in the last example above, where the wavelet function can be chosen differently depending on the scale of the randlet.

Because have compact support (or effectively compact support), they can emphasize large-scale and small-scale aspects of images, depending on the randlet scaling. Note that the functions above all have infinite support, but all are approximately zero at enough of a distance from the origin. Finally, randlets are normalized so that the inner product of a randlet with itself is 1.

Randlets of different types can be mixed together in the same basis. A basis with a single type of randlet is called a *pure* randlet basis, while a basis with multiple types of randlets is called a *mixed* randlet basis. The type of randlet or randlets used depends on the task. For image identification, Gaussian randlets perform the best. For edge detection, however, randlets with oscillations, such as Mexican Hat randlets or wavelet randlets, are better.

2.2 Generating a Randlet Transform

As was mentioned before, randlets are usually based on a handful of *mother randlets*, which generally are randomly scaled, rotated, and translated to form the randlet transform, although other operations can be performed as well, as long as the result is still perceptual. In practice, doing this as part of the randlet transform is very slow, so it is instead done ahead of time.

It is a simple matter to translate a randlet, so if there are multiple randlets in a basis that have the same rotation and scaling, only one copy of that randlet needs to be saved, and that copy can be translated across the image. However, if each randlet is scaled and rotated randomly, most randlets will be unique. The preprocessing phase will then become extremely slow, as almost all randlets must be scaled and rotated individually, and very memory-intensive, as each randlet must be stored in memory.

To avoid these problems the randlet basis is further constrained. Instead of allowing arbitrary rotations and scalings, a finite number of scalings and rotations are chosen for each mother randlet. These scalings and rotations can be chosen deterministically, or randomly based on the secret key.

A randlet basis is represented in an algorithm by a library of mother randlets at all allowed rotations and scalings, and a list of translations for the randlets from the library. Take the mother randlet $m(x, y)$. We will generate a library randlet by scaling by α horizontally and β vertically, and then rotating by θ degrees, to produce the library randlet $b[x, y]$.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad b[x, y] = m(x', y')$$

Now we will form the list of translations. Assume that the transform will be applied to images of height h pixels and width w pixels[2]. Let n be the size of the randlet basis and $F_K(\cdot)$ be an algorithm which selects a randlet from the library based on the secret key K. The list is generated by choosing randlets from the library using $F(\cdot)$, and translating those randlets uniformly at random $[0, 1, ...h]$ vertically and $[0, 1, ..., w]$ horizontally. If a randlet partially falls off the edge of the image, it is cropped to the edge of the image and renormalized during the transform and inverse transform.

2.3 Transform and Inverse Transform

This section describes the randlet transform and the inverse randlet transform as performed on an image of size $h \times w$. Section 2.3.2 describes how to apply the transform on images of different sizes.

The transform is done by projecting each randlet onto the image in turn. After each projection is taken, it is subtracted from the image. What remains of the image after the subtraction is called the *residual.* The transform continues in the manner, projecting each randlet onto the residual of the previous randlet, for all n randlets. When applying the randlet transform, it is important that the randlets be sorted approximately by the size of their support. The largest randlets should be projected and subtracted first, followed by the smaller randlets. Otherwise, the subtraction of the large randlets overwhelms the residual of the smaller randlets that were previously subtracted.

Transform coefficient c_j is computed by finding the inner product between randlet j and the residual of randlet $j - 1$. If the residual of randlet b_j is denoted by r_j, then

$$c_j = \sum_{y=1}^{h} \sum_{x=1}^{w} r_{j-1}[x, y] \cdot b_j[x, y] \qquad (1)$$

In practice, the coefficients are quantized. The quantized value is stored as the coefficient in the transform, and it is the quantized value that is used to compute the residual. If $Q(\cdot)$ is the quantizer, then

$$r_j[x, y] = r_{j-1}[x, y] - Q(c_j) \cdot b_j[x, y] \qquad (2)$$

[2]Section 2.3.2 describes how to apply the transform on images of different sizes.

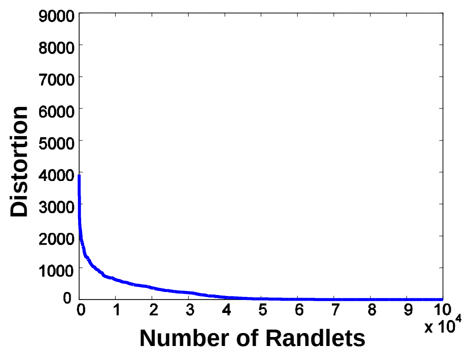


Figure 2: Convergence of the randlet transform for a sample image. The x-axis is the number of randlets that have been projected, the y-axis is distortion.

The inverse transform is extremely simple. Because residuals are used when taking the transform, the projection of each randlet is orthogonal to the previous randlets. Therefore, the original image can be reconstructed by simply adding together each randlet multiplied by its corresponding transform coefficient. If $\widetilde{I}[i,j]$ is the reconstructed image, then

$$\widetilde{I}[x,y] = \sum_{j=1}^{n} c_j \cdot b_j[x,y] \tag{3}$$

For both the transform and the inverse transform, the running time a basis of n randlets, each with area A, is $O(nA)$.

2.3.1 Practical Considerations

In application, the randlets are not projected across the entire image. The effective footprint of the randlets is computed in the preprocessing phase by considering only the area where the randlets have non-negligible value. The randlets are then "windowed" to this area.

Since the center of each randlet ranges uniformly across the whole image, the extremities of the randlets can reach outside of the image, which results in edge-effects. The edge effects can be eliminated by padding the image with a mirror-image of itself at each edge, and increasing h and w to accommodate the padding. In general, a padding of 5 to 10 pixels per side is sufficient.

After a number of projections have been taken, the residual often contains small details that are difficult for randlets to detect. Large randlets will overlap with these details, but will not be able to detect them because of the difference in scale. Smaller randlets could detect them, but will overlap them with very low probability. We have found that an effective way to solve this dilemma is to perturb each randlet and store the projection with the most power. This dramatically reduces the distortion for a given number of randlets.

Figure 2 shows, for a sample image, the distortion between the inverse transform and the image as a function of the number of randlets projected. The results are typical for the case when the randlets are perturbed. Preliminary, unoptimized tests have shown that when quantized and compressed, transform size is within a factor of two to four of JPEG compression.

2.3.2 Extending the Transform

A specific instance of the randlet transform is generated to work with a specific image size. In order to be useful, though, the randlet transform must be applicable to images of any size. There are two ways to achieve this.

First, the randlets can be chosen with centers in the range $[0, 1]$, and the transform can be expanded to the actual size of whatever image is chosen. In this method, the randlets themselves must also be scaled to match the size of the image. The advantage of this method is that many coefficients of the transform will become scaling-invariant. However, in order to ensure that this transform will scale well for large images, the randlet basis must be enormous. Many randlets will be redundant on small images, but are necessary to enable the transform to be taken to large scales.

The second method is to generate the transform for a relatively small image, say 128×128 or 256×256. Images larger than this size are decomposed into blocks and the transform is performed separately on all blocks. This is the preferred method, because it avoids the redundant basis functions of the previous two methods. However, the first methods can be useful for applications of the randlet transform such as hashing and watermarking, where a small number of basis functions are used.

In order to reduce edge effects between blocks, each block is padded as its transform is taken in exactly the same manner as the edge of images are padded in section 2.3.1. Furthermore, most images will not come in integer multiples of the block size. To deal with this, images are padded to achieve a size that is a multiple of the block size. This padding is removed as part of the inverse transform.

3 Image Identification and Hashing

An image identification system $H(\cdot)$ takes as input an image I and outputs a vector $v = H(I)$ which represents the image. Distances between vectors are computed as the Euclidian distance. v is computed by converting the image to a grayscale, scaling it to 256×256 pixels, and then computing the transform of the resulting image. We tested many randlet transform families and all of the wavelet transforms available in Matlab.

In an image identification system it is important that the identification vectors be kept secret, because an attacker with enough pairs (I_i, v_i) will be able to determine the basis functions that are being used, making it much easier to mount attacks.

For a randlet-based system, each image is transformed with a randlet transform, and v is assigned to be the transform coefficients. Residuals are not taken, so each randlet is projected directly onto the original image and randlets do not need to be sorted by size. If residuals were used, a change in an image that affected one coefficient would affect later coefficients not only through the change in the image, but also in the change in the residual. Furthermore, the power of later coefficients becomes small small when residuals are taken, making them less useful in identifying images. By not using residuals, each transform coefficient is made to depend only on the image. Since inversion is not necessary in image hashing is it possible to use relatively few basis functions to generate each image hash. 100 coefficients is sufficient for most purposes (see section 3.1).

For a DWT-based system, each image is transformed several times with the DWT, and the very lowest band is reshaped into a vector and assigned to v. It was determined experimentally that performance falls significantly when coefficients other than the low-low band are included in v. However, with each application of the DWT, the number of coefficients in the low-low band decreases by a factor of 4, which means that a 6-level

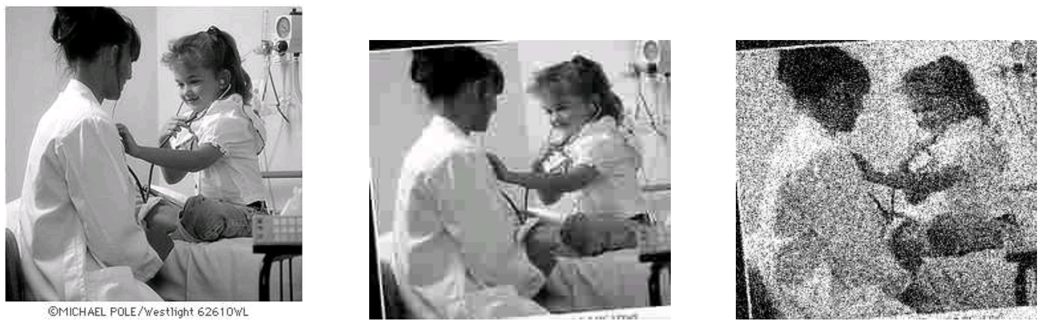


Figure 3: Illustration of the canonical attack. From left to right: the original image, the attacked image before adding Gaussian Noise, and the final attacked image

DWT transform would have only 16 coefficients. The randlet transform produces slightly superior performance to the 6-level DWT when only 16 randlet coefficients are used, but the randlet transform also has the option of increasing the number of coefficients to produce superior performance, an option the DWT does not have.

3.1 Experimental Results

All tests were performed on images taken from a library of 10,000 images. The images were of various sizes, but most were approximately 375×265 pixels in size. Unless otherwise stated, we ran our tests on 500 randomly-chosen images from the library. All of the wavelets present in the Matlab Wavelet Toolkit were tested. The Daubechies "db9" wavelet performed as well on image identification as any other wavelet tested, and we use it in all the DWT tests we present here.

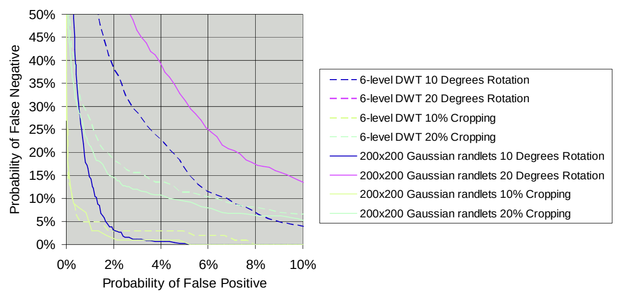
The top chart in figure 4 shows the results of cropping and rotating attacks. ROC curves are given for a 200×200 Gaussian randlet transform and for the 6-level DWT transform, both of which performed well for their respective transform types. Neither transform had any trouble with scaling (both enlarging and shrinking), additive Gaussian noise with standard deviations of 0.1 and 0.2 (pixels take values from 0 to 1), and JPEG compression of 50% and 5%. These attacks are not shown because they would not be visible as they are too close to the axes. The randlet transform significantly outperforms the DWT against cropping, and vastly outperforms the DWT against rotation attacks.

Note that these results do not hold true for all randlet transform and wavelet transforms. For example, DWT transforms using 5 or fewer levels do significantly worse in all tests, some other wavelet transforms perform very badly, and various randlet bases also perform poorly (see below).

To simplify and clarify the presentation, a strong, canonical composite attack was used for the results which follow. Unless otherwise stated, all ROC charts are illustrating image identification performance against this canonical attack. Figure 3 shows the effects of the attack on a sample image. Figure 4 contains a chart showing the ROC curves for various randlet and wavelet transforms against the canonical attack. The attack is, in order:

Rotation: Rotation of 5 degrees around a point 45% of the way from the top of the image and 45% of the way from the left of the image. Detecting a rotation is more difficult when the center of rotation is not the center of the image.

Cropping: The bottom of the image is cropped by 5% and the right edge by 4%. Cropping the image asymmetrically is a stronger attack than cropping symmetrically.



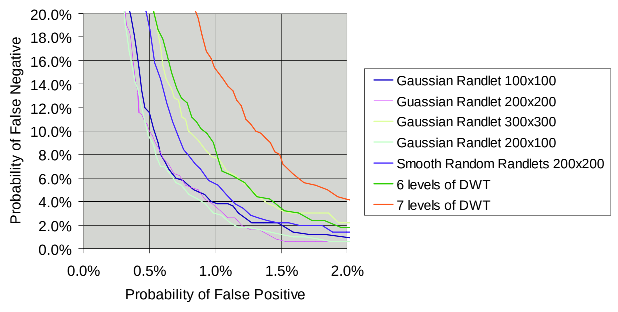


Figure 4: The chart on the top shows the ROC curves for one randlet transform and one wavelet transform against cropping and rotation attacks. The chart on the bottom shows the ROC curves for many different transforms against the canonical attack.

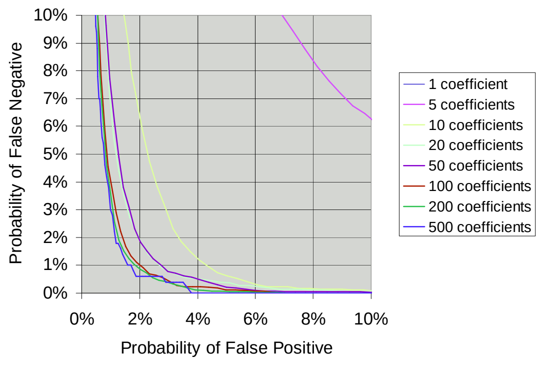
Scaling: The image is shrunk to 80% of its original width and 90% of its original height.

JPEG compression: JPEG compression with quality of 10 is applied.

Additive Gaussian Noise: Additive Gaussian noise with a standard deviation of 0.2 is added. Pixel values range from 0 to 1, and if the noise pushes a pixel below 0 or above 1, it is set to 0 or 1, respectively.

Random randlets, Mexican Hat randlets, and the DWT with 1 to 5 levels all perform very poorly. The 7-level DWT also performed poorly, although not as horribly as the previous examples. Next are 300×300 Gaussian randlets, smooth random randlets, and the 6-level DWT, which performed decently. The best performers, by far, are 100×100 200×100 and 200×200 Gaussian randlets.

Performance increases as the number of coefficients in a randlet transform increases,



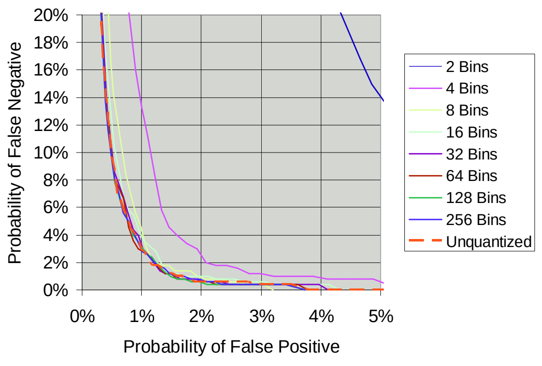


Figure 5: Both charts display the image identification performance of a 200×100 Gaussian randlet transform. The chart on the left shows how the number of coefficients affects the performance. Performance was so poor with a single coefficient that it does not even show up on the graph. The chart on the right shows the results of quantizing the transform coefficients with a non-uniform quantizer.

but there is a decreasing benefit to adding more coefficients, as can be seen in Figure 5. Using 5 or fewer coefficients gives poor performance under attacks. 10, 20, and 50 coefficients perform increasingly well, although not near the asymptotically optimal performance. 100 coefficient performs about as well as well as 200 coefficients, and both perform almost as well as 500 coefficients, which experimentally seems to be about the asymptotically optimal performance level. Against the composite attack, 100 coefficients is very close to optimal, and 200 coefficients give performance almost equivalent to optimal.

With 16 coefficients, a 200×100 Gaussian randlet transform performs slightly better than the 6-level DWT, which always has 16 coefficients. Not only does the randlet transform give better performance per coefficient than the DWT, it also has the ability to increase the number of coefficients, while the DWT does not.

3.1.1 Quantization

We tested a uniform quantizer over a range of quantization step sizes. A reasonable step size was one where the coefficients would be well distributed among various values. Quantization with a reasonable step size only slightly affected the performance of the randlet transform. Once the step size became so large that the coefficients were no longer well distributed, performance fell dramatically.

We also tested a non-uniform quantizer. It generated bins that were equiprobable over all transform coefficients in all un-attacked images, so that a coefficient randomly chosen from all images would be equally likely to fall in each bin. Of course, for a specific image the coefficients will be more likely to be in certain bins than others. For each bin, the quantization point was the mean value of all coefficients in the bin.

Figure 5 shows the effects of quantizing the transform coefficients with the non-uniform quantizer. The quantizer worked very well with the randlet transform. Performance suffered with too few bins, but with 8 bins performance was close to the unquantized performance, and with 16 bins it was almost equivalent. In these tests, therefore, 3 bits per coefficient is reasonable and 4 bits per coefficient achieves approximately the maximum performance possible with the randlet transform. Since 100 or 200 coefficients is sufficient, we can see that against the canonical attack 300 bits per image almost achieves optimal performance, and 800 bits per image achieves optimal performance.

When a transform is going to be used with an error-correcting code to form an image hash, it is desirable that as few coefficients as possible change after an attack. The more coefficients that change, the more robust the error-correcting code must be. Figure 6 illustrates the probability that a coefficient will be changed by our canonical attack for several different transforms. The randlet transform performs significantly better than the DWT.

3.1.2 Entropy

Figure 6 shows the entropy of transform coefficients for four transforms when taken over 500 images. All the coefficients for each transform were quantized with the non-uniform quantizer with various numbers of quantization bins. Then, each image was examined in turn and the entropy of each image's coefficients was calculated. The values shown in the chart are the entropy of each coefficient divided by the number of bits it would take to specify the coefficient's bin. We refer to this as the *normalized entropy.* The closer

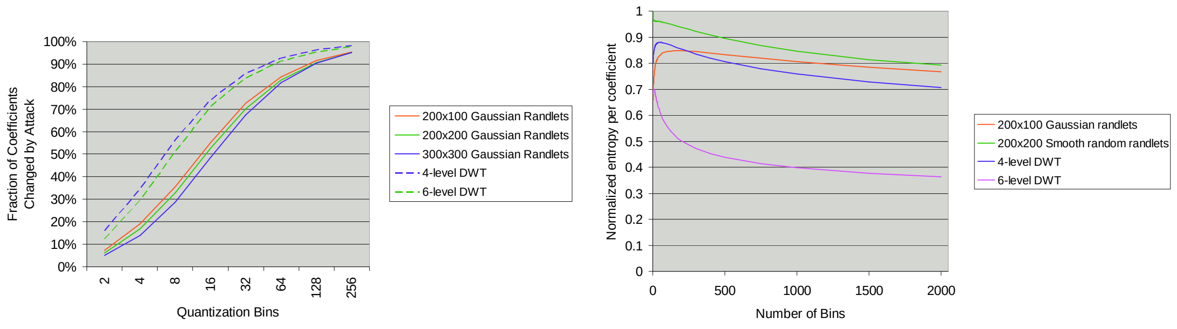


Figure 6: The chart on the left shows the fraction of coefficients that change when images are subjected to the canonical attack. The chart on the right shows the average entropy of transform coefficients for various transforms.

this value is to one, the closer the coefficients for that image are to being uniformly distributed across all bins, and the harder it is to predict the coefficients.

The 4-level DWT generates a reasonable amount of entropy. Unfortunately, it is very poor at image identification. The DWT with the best image identification performance was the 6-level DWT, but it is significantly worse than randlets at generating entropy in coefficients.

With the randlet transform, the entropy of the coefficients comes from both the distribution of images and the randomization of the transform. The randomization entropy is limited by the complexity of the image being transformed. For example, a completely blank image would not result in much entropy in the coefficients.

Because the entropy of coefficients depends on the complexity of the images being transformed, we tested the variance of images and subsets of images to determine if there was sufficient complexity in most images. 500 randomly chosen images were scaled to 256×256 pixels and decomposed into 200 randomly chosen, overlapping rectangular subsets of random size and position. The variance was calculated for the pixels in each rectangle, and a threshold was chosen so that over all rectangles from all images, a quarter of variances were less than the threshold and three quarters were greater than the threshold. Figure 7 shows that most images contain a significant amount of variance relative to this threshold. In our tests approximately 5% to 10% of images had low variance in many random rectangles. For example, an image of a far-away boat on the ocean could have very little variance in the water and the sky, and only have variance near the boat and the line of the horizon.

4 Conclusions

The key point of the randlet transform is that it is built with structured randomness; structured so as to give good performance, random so as to avoid worst-case performance. It may seem strange, for example, that the DWT performed reasonably well against many attacks but performed so poorly against rotation attacks, and one may suspect that the attacks were carefully chosen to produce the given results[3]. However, this just illustrates the point that randomization can help to avoid worst case performance. Evidently, rotation attacks induce worst case performance in the DWT.

We consider the randlet transform to be a new tool for signal processing. We demon-

[3]One would, of course, be incorrect in that suspicion. The attacks were chosen because they seemed like reasonable attacks, and were chosen in advance. All of the attacks that were tested are presented.

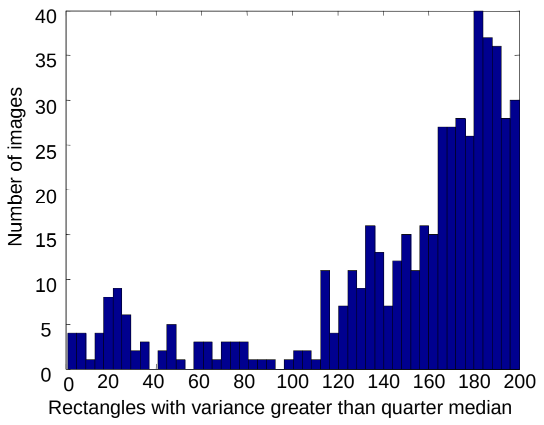


Figure 7: For each image, the number of rectangles exceeding the threshold was computed. This histogram shows how many images had each possible total. For example, at 100 on the x-axis the result is 2. That means that two images had exactly 100 rectangles with variance above the threshold and 100 rectangles with variance below the threshold. Note that most of the images are on the right of the graph, which corresponds to images with more complexity.

strate here its usefulness at image hashing and image identification, but we anticipate other applications as well. Future work will give a formal analysis of the randlet transform based on on finite distributions, as well as demonstrate applications of the randlet transform to watermarking and compression. We believe that many applications in signal processing can be enhanced by the explicit inclusion of randomness.

References

[1] E. Candès and D. Donoho. Curvelets: A surprisingly effective nonadaptive representation of objects with edges. Technical report, 1999.

[2] T. H. Cormen, E. Leiserson, Charles, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990. COR t 01:1 1.Ex.

[3] J. Fridrich. Key-dependent random image transforms and their applications in image watermarking.

[4] J. Fridrich. Visual hash for oblivious watermarking, 2000.

[5] R. M. F. i Ventura, P. Vandergheynst, and P. Frossard. Highly flexible image coding using non-linear representations. VCIP 2003, 2003.

[6] M. T. Malkin. *Applications of Randomness in Signal Processing*. PhD thesis, Stanford University, 2004 (estimated). in preparation.

[7] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[8] M. K. Mihcak and R. Venkatesan. New iterative geometric methods for robust perceptual image hashing.

[9] Z. D. N. Johnson and S. Jajodia. On “fingerprinting” images for recognition. 5th Int’l. Workshop on Multimedia Information Systems, 1999.

[10] R. Venkatesan, S. Koon, M. Jakubowski, and P. Moulin. Robust image hashing, 2000.