LOCAL RADIUS INDEX - A NEW TEXTURE SIMILARITY FEATURE

Yuanhao Zhai[1], *David L. Neuhoff*[1], *Thrasyvoulos N. Pappas*[2]

[1]EECS Dept., Univ. of Michigan, [2]EECS Dept., Northwestern Univ.

ABSTRACT

We develop a new type of statistical texture image feature, called a Local Radius Index (LRI), which can be used to quantify texture similarity based on human perception. Image similarity metrics based on LRI can be applied to image compression, identical texture retrieval and other related applications. LRI extracts texture features by using simple pixel value comparisons in space domain. Better performance can be achieved when LRI is combined with complementary texture features, e.g., Local Binary Patterns (LBP) and the proposed Subband Contrast Distribution. Compared with Structural Texture Similarity Metrics (STSIM), the LRI-based metrics achieve better retrieval performance with much less computation. Applied to the recently developed structurally lossless image coder, Matched Texture Coding, LRI enables similar performance while significantly accelerating the encoding.

***Index Terms*—** texture similarity, image coding, retrieval

1. INTRODUCTION

Texture similarity metrics play an important role in many image processing applications. Their goal is to make predictions consistent with human judgment. Due to the nature of texture, traditional point-by-point similarity metrics, such as MSE and PSNR, usually fail, for they are overly sensitive to small differences between images. Since textures have repetitive and statistical behaviors, it is natural that their similarity should be quantified with metrics based on statistics.

For different applications, texture similarity metrics need to possess different properties. For instance, texture classification needs semantically focused metrics that are shift and rotation invariant, e.g., metrics based on Local Binary Patterns (LBP) [1], and possibly even scale and viewpoint change invariant, e.g., [2–4]. However, in other problems such as image compression and some retrieval problems, the metric should be structural similarity focused, in that shifts can be tolerated but rotations, scalings and viewpoint changes, should be penalized monotonically [5–7]. In other retrieval problems, monotonicity is not required, and it is only necessary that the metric determine if two textures are sufficiently similar, or identical [6–10]. The Structural Texture Similarity Metrics STSIM1 [11] and STSIM2 [8, 12] are of this kind. In identical texture retrieval, STSIM2 has shown the best performance [8, 9]. When applied to image coding, satisfying results are obtained when STSIM2 is used in the recently developed structurally lossless image compression method, Matched Texture Coding (MTC) [13], which is a block-based image coder that uses the metric to decide if blocks of the image can be encoded by pointing to structurally similar ones in the already coded region.

In this paper, we focus on the need for texture metrics that assess the similarity of homogeneous textures in the context of image compression and retrieval where, as mentioned earlier, changes in rotations, scalings and viewpoints should be penalized monotonically. This is the domain for which STSIM type metrics have been developed [6, 8, 9, 14]. With this in mind, we propose a new type of statistical texture feature, called a Local Radius Index (LRI), and new texture similar metrics based on this feature in combination with other statistical features, such as LBP. These new metrics are computationally much simpler than STSIM metrics.

We test these new metrics on the problem of identical texture retrieval, as recently considered in [8, 9], and find that they outperform all previous metrics, while being computationally much simpler than the best of such. We also test one of these metrics in MTC. The results show it attains comparable compression and decoded image quality to that attained with STSIM2, but significantly accelerates the algorithm.

The rest of paper is organized as follows. The LRI feature is described in Sec. 2. And Sec. 3 presents new texture similarity metrics based on LRI and other features. Experimental results are discussed in Sec. 4, and Sec. 5 concludes the paper.

2. LOCAL RADIUS INDEX (LRI) FEATURES

Texture is a unique kind of visual signal and does not have a widely agreed definition. However, Portilla and Simoncelli [15] provide a general definition with which we agree: *"Loosely speaking, texture images are spatially homogeneous and consist of repeated elements, often subject to some randomization in their location, size, color, orientation, etc."* Generally speaking, a texture contains repetitive smooth regions and transition regions between them, i.e., edges. The sizes of the smooth regions influence the repetitive structure of the texture elements and can be captured by considering edges around them. The distance between two adjacent edges, which we will refer to as an *inter-edge distance*, is a good feature. Due to the stochasticity, the inter-edge distances are not constant, but there is a distribution of such. Also,

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

the inter-edge distance distributions may vary with angle and such variations can be key to characterizing the texture. Accordingly, one version of LRI is designed to capture inter-edge distance distributions in different directions. Similarly, another version of LRI is designed to capture the distributions of distances-to-nearest-edges. In particular, we propose two *LRI operators* that for each pixel output eight integer *directional indices*, each corresponding to one direction. Then for each direction, a histogram of the corresponding directional indices is computed. The collection of all eight histograms is considered an *LRI feature vector or statistic* for the image.

2.1. LRI-A

For each pixel, consider the eight directions corresponding to its eight nearest neighbors. For each direction, the LRI-A operator produces an integer index, depending on a *threshold* T and a *size limit* K. In particular, for the ith pixel and direction $d = 1$ to 8, the *directional index* $I_{i,d}$ is computed as follows:

1. $I_{i,d} = 0$ if the absolute difference between the current pixel and the adjacent one in direction d is less than T.
2. $I_{i,d} = \min\{j, K\}$ if $j > 0$ successive pixels in direction d are greater than the current pixel by at least T, and the $(j+1)$th pixel is not.
3. $I_{i,d} = \max\{-j, -K\}$ if $j > 0$ successive pixels in direction d are smaller than the current pixel by at least T, and the $(j+1)$th pixel is not.

Fig. 1 illustrates the LRI-A directional indices for several pixels. Note that $I_{i,d} = 0$ means pixel i is not adjacent to an edge in direction d, whereas $I_{i,d} \neq 0$ indicates the presence of an adjacent edge in direction d and $|I_{i,d}|$ represents the size of adjacent texture element. In addition, the sign of $(I_{i,d})$ indicates whether the edge is an increasing or decreasing one.

Of the two parameters involved, the threshold T determines what is considered an edge and also controls noise sensitivity. Large T makes the LRI-A operator noise insensitive and only sharp edges are detected. Small T has the opposite effect. T should be image dependent, and we have found that for homogeneous textures, choosing it to be proportional to the standard deviation of the image works well, e.g., T equals to the standard deviation divided by 2. The size limit K limits the maximum size of texture elements detected by LRI-A. In our testing we have found $K = 4$ is usually large enough.

As suggested earlier, the inter-edge distance distributions in each direction are captured by computing histograms of the directional indices in each direction. The *LRI feature vector* for an image then consists of these eight histograms.

2.2. LRI-D

Whereas LRI-A captures the width of the Adjacent smooth region, LRI-D captures the Distance to the nearest edge, i.e., to the boundary of the next smooth region. Specifically, the LRI-D operator calculates directional indices $I_{i,d}$ for the ith pixel in direction $d = 1$ to 8 as follows:

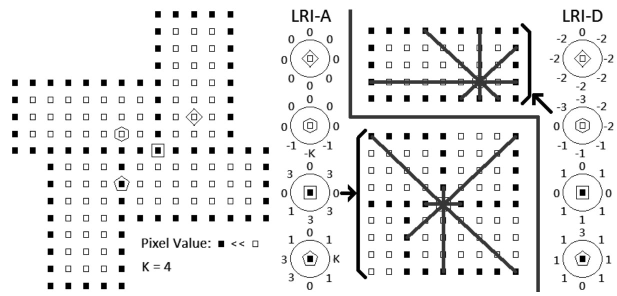

Fig. 1: Examples of LRI-A and LRI-D directional indices

1. $I_{i,d} = \min(j, K) \bmod K$, if for some $j \geq 1$, $j-1$ successive pixels in direction d have absolute difference with the current pixel less than T, and the jth pixel is greater than the current pixel by at least T.
2. $I_{i,d} = -(\min(j, K) \bmod K)$, if for some $j \geq 1$, $j-1$ successive pixels in direction d have absolute difference with the current pixel less than T, and jth pixel is smaller than the current pixel by at least T.

Note that $I_{i,d} \neq 0$ indicates the distance between the current pixel and the nearest edge in direction d is $|I_{i,d}|$ and the sign of $(I_{i,d})$ indicates whether the nearest edge is an increasing or decreasing one; $I_{i,d} = 0$ means the distance between the current pixel and the nearest edge in direction d is more than K. Fig. 1 illustrates the LRI-D directional indices for several pixels. The roles of T and K in LRI-D are similar to those in LRI-A. And as with LRI-A, the feature output is the collection of directional index histograms.

To accentuate the difference between LRI-A and LRI-D, consider their respective indices in direction d as one traverses a smooth region along a line with direction d from one of its boundaries to the next. Notice that the LRI-A indices are all zero, except for the last, corresponding to the pixel adjacent to the second boundary, whereas the LRI-D indices count down the distance to the second boundary. As a result, LRI-A indices are sparser, and LRI-D indices contain nonzero redundant information. Another viewpoint is that LRI-A focuses on representing size information of texture elements, whereas LRI-D concentrates on distance to adjacent texture elements.

2.3. Fast LRI algorithm

Computing the output of either LRI operator by directly using the definitions in Sec. 2.1 and 2.2 is fairly simple, as it only involves comparing the current pixel with its eight neighbors. Specifically, it requires up to $8K$ such comparisons per pixel. Nevertheless, it can be reduced to approximately $4K$ per pixel by exploiting the fact that the comparison between any pair of pixels is used twice, once for each pixel. In particular, for each of the four unsigned directions d, namely horizontal, vertical, diagonal and anti-diagonal directions, and each distance $k = 1$ to K, one can precalculate differences between all pairs of pixels separated by k in direction d and store such as *pixel difference images*. For instance, for the horizontal direction, one can calculate differences of every two pixels in the same line separated by k: $\Delta_{m,n}^k = x_{m,n} - x_{m,n+k}$, where for an $N \times N$ image, $1 \le m \le N$ and $1 \le n \le N - k$. Such computations can be viewed as filtering operations, with a total of $4K$ filters, namely, K in each of the four unsigned directions. For example, the magnitude of the frequency responses for the K filters in the horizontal direction has the form:

$$|H^k(\omega_X, \omega_Y)| = 2 - 2\cos(k\omega_Y),\ k = 1, 2..., K\,.$$

Although not orthogonal, such filters differ in orientation and scale. As we will see, the outputs of these filters can be also used to speed the computation of another useful feature.

3. LRI-BASED TEXTURE SIMILARITY METRICS

This section first describes a texture similarity metric based purely on an LRI feature, and then proposes better metrics that combine LRI with complementary texture features.

3.1. A Purely LRI-based Similarity Metric

A similarity metric based purely on LRI can be easily derived. First concatenate the eight histograms to form an $8\times(2K+1)$ dimensional vector and normalize it to unit length. Then compare the similarity of the vectors obtained from two different images using information theoretic divergence [16]. The resulting similarity score has non-negative values, with 0 meaning identical, and a small metric value meaning similar.

3.2. Local Binary Patterns (LBP)

LBP [1] is a widely used feature in texture analysis. Whereas LRI extracts features in eight radial directions without considering their correlations, LBP, on the contrary, detects patterns in the tangential direction, which is complementary to LRI. Hence, better metric performance can be expected if LRI and LBP are combined. Unlike the original version of LBP, we propose to compute LBP using the surrounding 8 pixels without doing interpolation to estimate image values on a circle surrounding the current pixel. From experiments, we found that such interpolation does not improve overall performance. Since pixel differences are already computed when LRI is computed, when interpolation is not required, virtually no additional computation is needed to compute LBP indices.

3.3. Subband Contrast Distribution (SCD)

Subband decompositions oriented to visual perception have been quite central to CW-SSIM [17] and STSIM [8, 11, 12] metrics. We find that the variances in each subband, which indicate subband contrasts, suffices to provide useful complementary statistics to LRI, which by itself is contrast invariant.

Specifically, we propose two closely related features that we collectively refer to as *Subband Contrast Distribution*. The first applies a *real* steerable pyramid decomposition [15] with 3 scales and 4 orientations, similar to what is used in CW-SSIM and STSIM. The resulting feature is the vector $(\sigma_1^2, \dots, \sigma_{12}^2)$ of 12 subband variances. The similarity of the SCD vectors from two different images, x and y, is assessed using the approach taken in SSIM type metrics [17, 18] :

$$\text{SCD}(x, y) = \prod_{r=1}^{12} \frac{2\sigma_{x,r}\sigma_{y,r} + C}{\sigma_{x,r}^2 + \sigma_{y,r}^2 + C}$$

where $\sigma_{x,r}^2, \sigma_{y,r}^2$ denotes the variance of the rth subband of x and y, respectively. As such, SCD(x, y) ranges from 0 to 1, with 1 indicating identical. Multiplicatively combining all terms has the interpretation that only when all subbands have similar contrasts can SCD have a high value. C is a small positive constant that prevents the denominator from being zero and improves robustness when subbands have small energy. Typically, $C = 10$, when images take values from 0 to 255.

The second approach makes use of the pixel difference images, produced by the fast LRI algorithm. As mentioned previously, these are filtered versions of an image, with differing orientations and scales. As such, they can be used in place of the steerable pyramid decomposition. The statistic formation and similarity scoring formula are the same as before. To distinguish this approach from the previous, it will be called estimated SCD or SCD_{EST}; it needs no significant additional computations beyond those required for LRI.

3.4. Intensity Penalization (IP)

Besides texture patterns and contrast, intensity can also influence human similarity judgment. To penalize large intensity difference, we propose the following penalization function:

$$\text{IP}(x, y) = \Big[\max\big(T_L, |I_x - I_y|\big)/256\Big]^{p_L}, T_L = 10, p_L = 2,$$

where I_x and I_y are mean values of images x and y, T_L serves as a kind of just noticeable intensity difference, and p_L determines the severity of penalization. The denominator 256 is chosen presuming images with pixel intensities ranging from 0 to 255. A smaller value would cause less penalization.

3.5. LRI-based Similarity Metric

After experimenting with a number of ways of combining LRI with complementary features, we propose the following metric combining LRI with LBP, SCD (or SCD_{EST}) and IP:

$$\text{LRI}^+ = \text{LRI}^{p_1} \times \text{LBP}^{p_2} \times f(1 - \text{SCD})^{p_3} \times \text{IP}\,,$$

where LRI^+ is the metric value and $f(x)$ is an increasing function of x on $[0, 1]$. We use $f(x) = \tan(x\frac{\pi}{2})$, but also experimented with $f(x) = x$. Parameters p_1, p_2, p_3 weight the different features, with typical values 1, 1.1 and 1.2.

LRI^+ is nonnegative, with 0 meaning identical, and a small metric value meaning similar. Since the values of LRI^+ are often close to 0, for convenience, we usually report the logarithm of LRI^+. As a benchmark, we have found that $\log_{10} \text{LRI}^+ = -6$ indicates very similar textures.

Computational complexity

An LRI-based metric requires many fewer computations than STSIM2. Leaving aside transform costs, STSIM2 (with a global window) requires approximately 500 operations per pixel, whereas LRI and SCD require 16 and 36, respectively. Of course both STSIM2 and SCD require the steerable pyramid decomposition, the computation of which requires 14 FFT, each requiring $5 \log N$ operations per pixel for images with N pixels [19]. However, when $\mathrm{SCD}_{\mathrm{EST}}$ is applied instead of SCD, such transform costs can be reduced.

4. TESTS OF THE METRICS

4.1. Identical Texture Retrieval

One important test of a metric is its ability to distinguish identical from nonidentical textures. To make such a test we adopt the identical texture retrieval setup of [8, 9]. This includes a database of 1181 images carefully cropped from 485 photographs [20], chosen to have homogeneous textures. Images are considered to be identical if and only if they were cropped from the same photograph. One benefit of this setup is that subjective experiment is not needed to establish ground truth.

For a given metric, the retrieval test is conducted as follows. Each image in the database serves as a query, and for each query, all other images are ordered according to their similarity to the query, as measured by the given metric. The goodness of the metric is then judged by how high in the ordering the images identical to the query typically appear. As in [8, 9], we consider four retrieval performance measures: Precision at 1 (P@1) [8], Mean Reciprocal Rank (MRR) [21], Mean Average Precision (MAP) [22] and the Area Under the Receiver Operating Characteristic (AUROC) [8]. While the first three measure the ability of the metric to correctly rank similarity of other images to a query, they do not test whether a metric value indicating identicality of a texture y to a texture x is the same for all x. The latter is tested by AUROC.

Test results are shown in Table 1 for several LRI-based metrics, as well as the results reported in [8] for the following metrics: SSIM [18], CW-SSIM [17], LBP [1] and STSIM2 [8, 12]. The LRI-based metrics include, LRI-A, LRI-D, LRI_a^+ (LRI-A, LBP, SCD, IP), LRI_b^+ (LRI-A, LBP, $\mathrm{SCD}_{\mathrm{EST}}$, IP) and LRI_c^+ (LRI-D, LBP, SCD, IP). The results show that the three versions of LRI^+ perform better than all other metrics, with STSIM2 not far behind. It is interesting that while LRI-D by itself is not as good as LRI-A by itself, when combined with the other features, it works just as well, as the others compensate for its shortcomings. It is also interesting that the performance of the LRI_b^+ is so similar to that of LRI_a^+ and LRI_c^+. This shows that substituting the low complexity $\mathrm{SCD}_{\mathrm{EST}}$ for SCD has little effect on performance. Also note that LRI-A, LRI-D and SCD are surprisingly good in AUROC.

4.2. Structurally Lossless Image Coding Experiment

Another test of a metric is its ability to be used in MTC [13] to judge if an image block can be satisfactorily coded by pointing to some candidate block from the already coded portion of the image. Accordingly, we found that MTC performance

Table 1: Identical texture retrieval tests.

Metrics	P@1	MRR	MAP	AUROC
SSIM [8]	9%	11%	6%	0.45
CW-SSIM [8]	39%	46%	40%	0.92
LBP [8]	90%	92%	86%	0.59
STSIM2 [8]	96%	97%	92%	0.98
LRI-A	91.8%	93.5%	86.9%	0.982
LRI-D	83.2%	86.9%	78.8%	0.974
SCD	83.7%	88.0%	80.5%	0.984
LRI_a^+	**98.7%**	**99.2%**	**96.4%**	**0.994**
LRI_b^+	**98.1%**	**98.7%**	**95.4%**	**0.994**
LRI_c^+	**99.0%**	**99.2%**	**96.3%**	**0.994**

was at least as good with LRI_b^+ as with STSIM2. For example, Fig. 2 shows the "waterfall" image MTC coded at a very low coding rate, 0.18 bpp, with both metrics. One can see that the two metrics result in very similar quality. However, the latter runs 6.3 times faster for this example. As an additional benchmark Fig. 2 also shows the image coded with JPEG at the same rate. For much of the image, JPEG evidences blocking artifacts and MTC is clearly better. However, MTC has a few flaws, such as eliminating one or two "floats" in the water.

5. CONCLUSION

A new type of texture feature, LRI, and metrics based on it are proposed. LRI is a simple to compute feature that enables the improvement of texture similarity metrics. Compared to STSIM2, the LRI-based metrics are much simpler with better performance when tested on the same database. And when used in MTC, an LRI-based metric obtains similar performance while significantly accelerating the encoding.

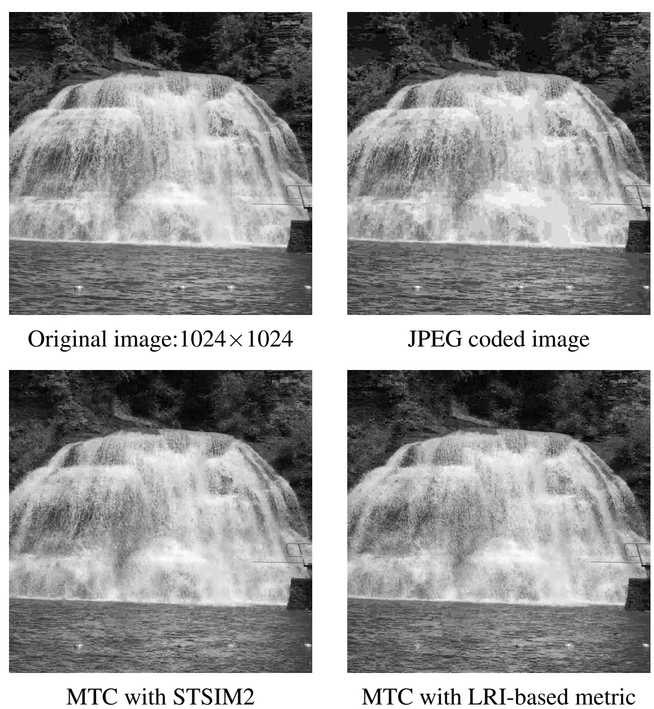

Fig. 2: Example of "waterfall" image coded at 0.18 bpp

6. REFERENCES

[1] T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 971-987, Jul. 2002.

[2] S. Lazebnik, C. Schmid and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, pp. 1265-1278, Aug. 2005.

[3] M. Mellor, B.-W. Hong and M. Brady, "Locally rotation, contrast, and scale invariant descriptors for texture analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 52-61, Jan. 2008.

[4] G.-S. Xia, J. Delon and Y. Gousseau, "Shape-based invariant texture indexing," *Int. J. Comput. Vis.*, DOI 10.1007/s11263-009-0312, Nov. 2009.

[5] J. Zujovic, T.N. Pappas, D.L. Neuhoff, R. van Egmond, and H. de Ridder, "Subjective and objective texture similarity for image compression," *ICASSP*, pp. 1369-1372, Kyoto, Mar. 2012.

[6] J. Zujovic, T.N. Pappas, D.L. Neuhoff, R. van Egmond and H. de Ridder, "Effective and efficient subjective testing of texture similarity metrics," submitted to *IEEE Trans. Image Proc.*, Sept. 2012.

[7] J. Zujovic, *Perceptual Texture Similarity Metrics*, Ph.D. Thesis, Northwestern University, 2011.

[8] J. Zujovic, T.N. Pappas and D.L. Neuhoff, "Structural texture similarity metrics for image analysis and retrieval," to appear in *IEEE Trans. Image Proc.*, Apr. 2012.

[9] J. Zujovic, T.N. Pappas and D.L. Neuhoff, "Perceptual similarity metrics for retrieval of natural textures," *IEEE Int. Workshop Multimedia Sig. Proc. (MMSP)*, Rio De Janeiro, pp. 1-5, Oct. 2009.

[10] J. Zujovic, T.N. Pappas, D.L. Neuhoff, R. van Egmond and H. de Ridder, "A new subjective procedure for evaluation and development of texture similarity metrics," *Proc. IEEE 10th IVMSP Workshop*, pp. 123-128, June 2011.

[11] X. Zhao, M.G. Reyes, T.N. Pappas and D.L. Neuhoff, "Structural texture similarity metrics for retrieval applications," *IEEE Intl. Conf. on Image Proc. (ICIP)*, pp. 1196-1199, Oct. 2008.

[12] J. Zujovic, T.N. Pappas and D.L. Neuhoff, "Structural similarity metrics for texture analysis and retrieval," *IEEE Intl. Conf. on Image Proc. (ICIP)*, pp. 2225-2228, Nov. 2009.

[13] G. Jin, Y. Zhai, T.N. Pappas and D.L. Neuhoff, "Matched-texture coding for structurally lossless compression," *IEEE Intl. Conf. on Image Proc. (ICIP)*, Oct. 2012.

[14] T.N. Pappas, J. Zujovic and D.L. Neuhoff, "Image analysis and compression: renewed focus on texture," *Vis. Inf. Proc. Comm.*, Proc. SPIE vol. 7543, pp. 75430N-1-12, Jan. 2010.

[15] J. Portilla and E.P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *Int. J. Comput. Vis.*, 40: 49-70, Oct. 2000.

[16] S. Kullback and R.A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, pp. 79-86, Mar. 1951.

[17] M.P. Sampat, Z. Wang, S. Gupta, A.C. Bovik and M.K. Markey, "Complex wavelet structural similarity: a new image similarity index," *IEEE Trans. Image Proc.*, vol. 18, pp. 2385-2401, Nov. 2009.

[18] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Proc.*, vol. 13, pp. 600-612, Apr. 2004.

[19] J. Cooley and J. Tukey, "An algorithm for the machine calculation of complex fourier series," *Math. Comput.*, 19: 297-301, 1965.

[20] "Corbis stock photography," http://www.corbis.com.

[21] E.M. Voorhees, "The trec-8 question answering track report," *Proc. of TREC-8*, pp. 77-82, 1999.

[22] E.M. Voorhees, "Variations in relevance judgments and the measurement of retrieval effectiveness," *Inform. Process. Manag.*, vol. 36, pp. 697-716, Sept. 2000.