

# A 2D TEXTURE IMAGE RETRIEVAL TECHNIQUE BASED ON TEXTURE ENERGY FILTERS

Motofumi T. Suzuki, Yoshitomo Yaginuma and Haruo Kodama

*National Institute of Multimedia Education, 2-12 Wakaba, Mihama-ku, Chiba-shi 261-0014, Japan*

**Keywords:** Laws' filters, Convolution, 2D Image Database, Image Retrieval, Shape Features.

**Abstract:** In this paper, a database of texture images is analyzed by the Laws' texture energy measure technique. The Laws' technique has been used in a number of fields, such as computer vision and pattern recognition. Although most applications use Laws' convolution filters with sizes of  $3 \times 3$  and  $5 \times 5$  for extracting image features, our experimental system uses extended resolutions of filters with sizes of  $7 \times 7$  and  $9 \times 9$ . The use of multiple resolutions of filters makes it possible to extract various image features from 2D texture images of a database. In our study, the extracted image features were selected based on statistical analysis, and the analysis results were used for determining which resolutions of features were dominant to classify texture images. A texture energy computation technique was implemented for an experimental texture image retrieval system. Our preliminary experiments showed that the system can classify certain texture images based on texture features, and also it can retrieve texture images reflecting texture pattern similarities.

## 1 INTRODUCTION

In the 1990s, intensive research was conducted on content-based image retrieval methods. Unlike traditional keyword-based image retrieval methods, content-based image retrieval methods have used image features as indices for image databases. The image features describe characteristics of the images which include colors, shapes and textures. Since image feature indices can be extracted automatically by using software programs, this process is more efficient and faster compared to that of keyword indices which are assigned by human hands. Various image feature extraction techniques have been proposed based on computer algorithms, such as co-occurrence matrices (Haralick et al., 1973), Markov random field modeling (Cross and Jain, 1983), Gabor filtering (Manjunath and Ma, 1996), Wavelet (Chang and Kuo, 1993), and HLAC (Kato, 1992). Also, various statistical learning techniques have been proposed to improve retrieval rates by learning from the sample image data (Kato, 1992). Such techniques include use of self-organizing maps (SOM), neural networks (NN) and support vector machines (SVM). Details of image features and a related retrieval technique survey can be found in several papers (Veltkamp and Tanase, 2000) (Datta et al., 2008).

In our experiments, the famous Laws' texture energy measure technique is applied to a 2D texture image database. Although a typical application uses Laws' filters of  $3 \times 3$  and  $5 \times 5$ , the sizes of the filters were extended to  $7 \times 7$  and  $9 \times 9$  in our experiments. These multiple resolutions of the filters were used for extracting image features from images of a texture database. The extracted features were analyzed by linear discriminant analysis to determine which resolutions of features were best suited for classifying the texture database. Also, principal component analysis (PCA), and the k-nearest neighbor (KNN) algorithm was used for the similarity retrieval of a texture database.

## 2 TEXTURE ENERGY FILTERS

In this section, (1) Laws' texture energy measures, (2) convolution and (3) rotation invariant features are discussed.

### 2.1 Laws' Texture Energy Measures

The texture analysis technique based on the texture energy measure which was developed by K. I. Laws

(Laws, 1979) (Laws, 1980) has been used for many applications of image analysis for classification and segmentation. The texture energy measurements for 2D images are computed by applying convolution filters. In the technique, three basic filters were used as follows:

$$\begin{aligned} L3 &= (1\ 2\ 1) \\ E3 &= (-1\ 0\ 1) \\ S3 &= (-1\ 2\ -1) \end{aligned}$$

The initial letters of these filters indicate Local average (or Level), Edge detection, and Spot detection. The numbers followed by the initial letters indicate lengths of the filters. In this case, the length of the filters is three. Often, extended lengths of filters are used for 2D image analysis. The extension of the filters can be done by convolving the pairs of these filters together. For example, filters with a length of five can be obtained by convolving pairs of filters with a length of three. In this convolution process, nine filters ( $3 \times 3$ ) can be formed, and five of them are distinct. The following is a set of one dimensional convolution filters of a length of five:

$$\begin{aligned} L5 &= (1\ 4\ 6\ 4\ 1) \\ E5 &= (-1\ -2\ 0\ 2\ 1) \\ S5 &= (-1\ 0\ 2\ 0\ -1) \\ W5 &= (-1\ 2\ 0\ -2\ 1) \\ R5 &= (1\ -4\ 6\ -4\ 1) \end{aligned}$$

The initial letters of these filters stand for Local average (or Level), Edge, Spot, Wave, and Ripple. All filters are zero-sum filters except for the *L5* filter. Many applications use Laws' filter with a size of 3 and 5 for extracting texture energy values. In our experiments, we have extended the filter sizes to 7 and 9. The filters of 7 can be obtained by convolving filters of a length of five and filters of a length of three as follows:

$$\begin{aligned} Xa7 &= (1, 6, 15, 20, 15, 6, 1) \\ Xb7 &= (1, 4, 5, 0, -5, -4, -1) \\ Xc7 &= (-1, -2, 1, 4, 1, -2, -1) \\ Xd7 &= (1, 0, -3, 0, 3, 0, -1) \\ Xe7 &= (1, -2, -1, 4, -1, -2, 1) \\ Xf7 &= (1, -4, 5, 0, -5, 4, -1) \\ Xg7 &= (-1, 6, -15, 20, -15, 6, -1) \end{aligned}$$

By using a similar approach, one dimensional kernels of a length of nine are obtained as follows:

All the kernels are zero-sum kernels except for *Xa7* and *Ya9*. Simple sequential labels *X* and *Y* were assigned for the filters 7 and 9 for convenience, although more meaningful labels such as *L*, *E*, *S*, *W*

$$\begin{aligned} Ya9 &= (1, 8, 28, 56, 70, 56, 28, 8, 1) \\ Yb9 &= (1, 6, 14, 14, 0, -14, -14, -6, -1) \\ Yc9 &= (-1, -4, -4, 4, 10, 4, -4, -4, -1) \\ Yd9 &= (1, 0, -4, 0, 6, 0, -4, 0, 1) \\ Ye9 &= (1, 2, -2, -6, 0, 6, 2, -2, -1) \\ Yf9 &= (-1, 2, 2, -6, 0, 6, -2, -2, 1) \\ Yg9 &= (-1, 4, -4, -4, 10, -4, -4, 4, -1) \\ Yh9 &= (1, -8, 28, -56, 70, -56, 28, -8, 1) \\ Yi9 &= (1, -6, 14, -14, 0, 14, -14, 6, -1) \end{aligned}$$

and *R* can be used. (Obviously, *Xa7* can be labeled *L7*, and *Ya9* can be labeled *L9*.)

These one dimensional filters are used to generate two dimensional filters by combining these one dimensional filters. The set of two dimensional filters with lengths of three ( $3 \times 3$ ) are as follows:

$$\begin{aligned} L3L3, L3E3, L3S3 \\ E3L3, E3E3, E3S3 \\ S3L3, S3E3, S3S3 \end{aligned}$$

In a similar manner, two dimensional filters with the lengths of  $5 \times 5$ ,  $7 \times 7$  and  $9 \times 9$  can be obtained. Furthermore, three dimensional filters such as  $3 \times 3 \times 3$  can be generated by combining basic one dimensional filters (Suzuki and Yaginuma, 2007).

## 2.2 Convolution

Once the two dimensional filters are obtained, these filters are used to convolve the 2D texture image. The convolution of image *I* and filter *F* with a size of  $2t + 1$  by  $2t + 1$  is expressed by the following equation:

$$R(i, j) = F(i, j) * I(i, j) = \sum_{k=-t}^t \sum_{l=-t}^t F(k, l) I(i+k, j+l) \quad (1)$$

where '\*' denotes two dimensional convolution computation. For the next step, the windowing process is applied to convolved images. In this process, texture energy values are computed. Every pixel in the convolved images is replaced with a texture energy measure value at the pixel. In the Laws' paper, a  $15 \times 15$  square around each pixel is added together with the values of the neighborhood pixels. In this computation, Laws introduced "squared magnitudes" and "absolute magnitudes" to compute texture energy (Laws, 1979) (Laws, 1980). For considering computation efficiency, "absolute magnitude" is used in general. This computation process can be expressed by the following equation:

$$E(l, m) = \sum_{i=l-t}^{l+t} \sum_{j=m-t}^{m+t} |K(i, j)| \quad (2)$$

where  $K$  is the local features, and it is smoothed at position  $(l, m)$  by using a  $(2t + 1) \times (2t + 1)$  window.

### 2.3 Rotation Invariant Features

Some filters are identical if they are rotated 90 degrees. The features which are computed from the filters can be combined as similar features, and these features are treated as rotation invariant features in the order of 90 degrees. For example, feature  $E5L5$  can be combined with  $L5E5$ , and newly created features are denoted as  $E5L5_R$  where  $R$  means the "rotation invariant" feature. On the other hand, some features such as  $E5E5$ ,  $S5S5$ ,  $W5W5$  and  $R5R5$  can not be combined. Since the rotation invariant features were combined with two features, the feature values were scaled by 2. Therefore, the rotation invariant features are divided by 2 for the purpose of normalization. Figure 1 shows rotation invariant features for filter lengths of  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$  and  $9 \times 9$ . (In this paper, rotation invariant features are denoted as  $3 \times 3_R$ ,  $5 \times 5_R$ ,  $7 \times 7_R$  and  $9 \times 9_R$ , respectively.) Also the number of corresponding rotation invariant features is shown in Figure 1. The number of rotation invariant features  $F_r$  can be computed by equation  $F_r = n^2 - ((n^2 - n)/2)$  where  $n$  is filter size.

R.	Rotation Invariant Features	Num.
$3^2$	$L3L3, L3E3_R, L3S3_R, E3E3$ $E3S3_R, S3S3$	$9 \rightarrow 6$
$5^2$	$L5L5, L5E5_R, L5S5_R, L5R5_R$ $L5W5_R, E5E5, E5S5_R, E5R5_R$ $E5W5_R, S5S5, S5R5_R, S5W5_R$ $R5R5, R5W5_R, W5W5$	$25 \rightarrow 18$
$7^2$	$XaXa7, XaXb7_R, XaXc7_R, XaXd7_R$ $XaXe7_R, XaXf7_R, XaXg7_R, XaXb7,$ $XbXc7_R, XbXd7_R, XbXe7_R, XbXf7_R$ $XbXg7_R, XcXc7, XcXd7_R, XcXe7_R$ $XcXf7_R, XcXg7_R, XdXd7, XdXe7_R$ $XdXf7_R, XdXg7_R, XeXe7, XeXf7_R$ $XeXg7_R, XfXf7, XfXg7_R, XgXg7$	$49 \rightarrow 28$
$9^2$	$YaYa9, YaYb9_R, YaYc9_R, YaYd9_R$ $YaYe9_R, YaYf9_R, YaYg9_R, YaYh9_R$ $YaYi9_R, YbYb9, YbYc9_R, YbYd9_R$ $YbYe9_R, YbYf9_R, YbYg9_R, YbYh9_R$ $YbYi9_R, YcYc9, YcYd9_R, YcYe9_R$ $YcYf9_R, YcYg9_R, YcYh9_R, YcYi9_R$ $YdYd9, YdYe9_R, YdYf9_R, YdYg9_R$ $YdYh9_R, YdYi9_R, YeYe9, YeYf9_R,$ $YeYg9_R, YeYh9_R, YeYi9_R, YfYf9$ $YfYg9_R, YfYh9_R, YfYi9_R, YgYg9$ $YgYh9_R, YgYi9_R, YhYh9, YhYi9_R$ $YiYi9$	$81 \rightarrow 45$

Figure 1: Rotation invariant features ( $3 \times 3_R$ ,  $5 \times 5_R$ ,  $7 \times 7_R$  and  $9 \times 9_R$ ).

## 3 EXPERIMENTS AND RESULTS

This section describes (1) experimental textures, (2) comparison of filters for various lengths, (3) similarity retrievals of a texture database

### 3.1 Experimental Textures

In our experiment, a portion of the OUTEX (Ojala et al., 2002) texture database was used. It contains 6380 textures (Outex-TR-00000 data set). It consists of 319 classes of textures with 20 textures in each class. The database contains both macro-textures and micro-textures. Texture sizes are  $128 \times 128$  in RAS image data format. In our experiments, these data are converted to sizes of  $128 \times 128$  images in PGM format with 65536 grey scale colors.

### 3.2 Comparison of Filters for Various Lengths

Various lengths of Laws' filters were examined to determine whether the filters can classify the textures. For the experiments, five classes of textures were selected from a portion of the OUTEX database. The examples of the textures are shown in Figure 2. Each class contains 20 textures, thus there were a total of 100 textures. Figure 3 shows examples of convolved texture images (seeds textures) using Laws' filters ( $3 \times 3$ ,  $5 \times 5$  and  $7 \times 7$ ). As the filter sizes are increased, the resolutions of the convolved images become coarser. Texture images analyzed by large size filters such as  $7 \times 7$  contain very similar convolved texture images. Some features associated with these similar convolved images are considered redundant, and these convolved images can be eliminated for efficient computation. The elimination of the redundant features avoids a bias from the features due to dimensionality.

Once the entire texture data were analyzed by Laws' filters, the texture energy could be computed by the technique mentioned in the previous section (Section 2). For the texture energy computation,  $15 \times 15$  smoothing windows were used for the experiments. These texture energy values and texture class identification numbers were used as input of a discriminant analysis (LDA). In this experiment, Laws' filters of sizes  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$  and  $9 \times 9$  were examined. For each filter size, rotation invariant (mentioned in section 2.3) features ( $3 \times 3_R$ ,  $5 \times 5_R$ ,  $7 \times 7_R$  and  $9 \times 9_R$ ) were also computed. Ten textures for each class were used for the learning data sets (10 textures  $\times$  5 classes).

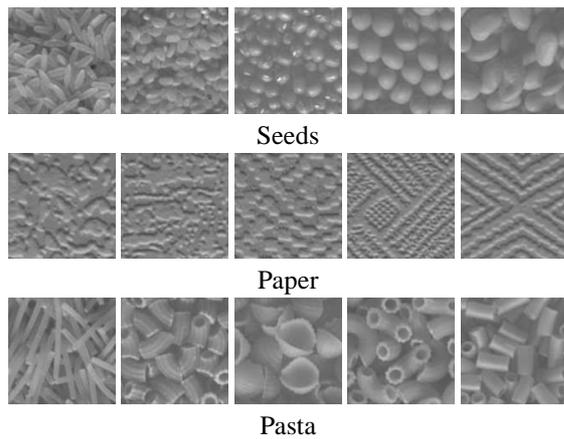


Figure 2: Example of textures.

Figure 4 shows texture data of discriminant coordinates for features based on the Laws  $7 \times 7$  filter. Four discriminant functions are labeled  $LD1$ ,  $LD2$ ,  $LD3$  and  $LD4$  in the figure. The labels  $a$ ,  $b$ ,  $c$ ,  $d$  and  $e$  in the figure indicate each class of textures. When each function has good discriminating power, there is no overlap observed between texture classes.

Figure 5 shows proportions of eigenvalues for the discriminant functions. The eigenvalues reflect the amount of variance explained in the grouping variables by the predictors. In the figure, the proportion of the eigenvalues estimates the relative importance of the discriminant functions. In the case of the figure, discriminant functions  $LD1$ ,  $LD2$ , and  $LD3$  have more discriminating power compared to that of  $LD4$ .

The larger the coefficient values of a predictor in the discriminant function, the more important its role in the discriminant function. Therefore, these predictors (image features) associated with the larger coefficients are influential predictors (image features) for classifying textures. In this example, features  $XeXe7$ ,  $XfXd7$  and  $XfXf7$  had relatively large coefficients for the  $LD1$  function, and they are considered as important image features for classifying our experimental texture dataset. Important image features vary depending on the patterns which are contained in the texture dataset.

Figure 6 shows a set of test texture data (textures of seeds) classified by the discriminant functions trained by the training data set. In the figures, rows represent predicted texture types, and columns represent actual texture types. For the  $5 \times 5$  filter case, there are 4 textures misclassified. Two percent ( $1/50$ ) of ' $a$ ' textures are misclassified as ' $f$ ' textures, 2% ( $1/50$ ) of ' $b$ ' textures are misclassified as ' $e$ ', 2% ( $1/50$ ) of ' $d$ ' textures are misclassified as ' $c$ ', and 2% ( $1/50$ ) of ' $e$ ' textures are misclassified as ' $d$ ', Therefore, 92% (46 out of 50) of test textures are classified

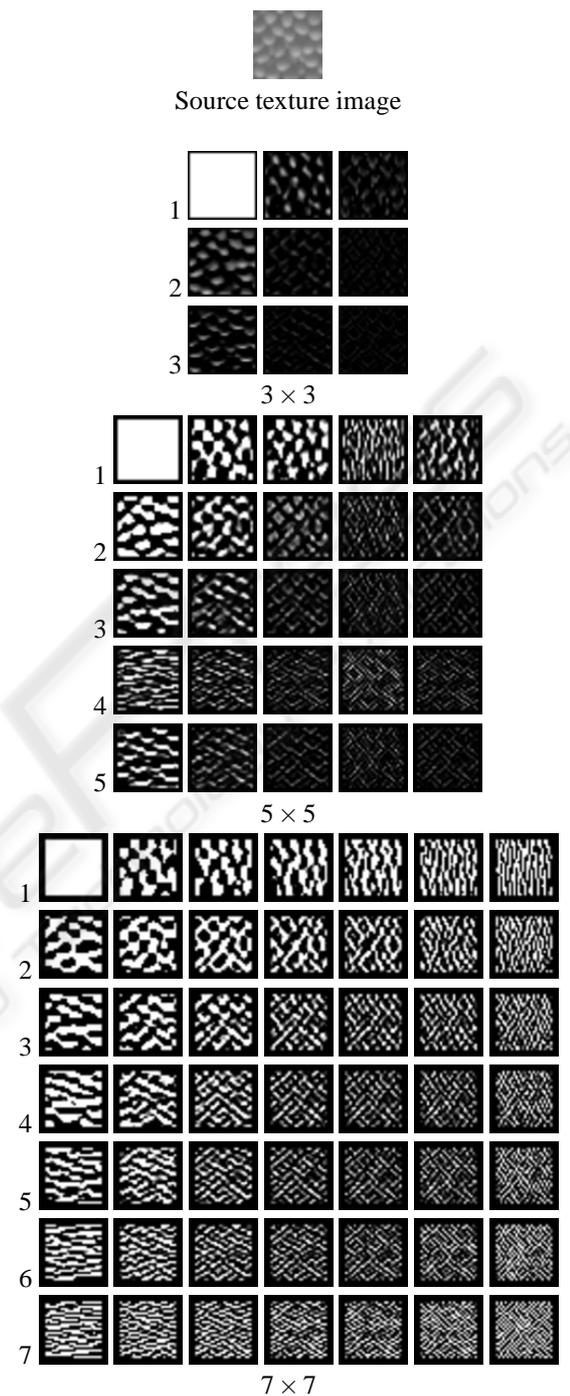


Figure 3: Examples of convolved texture images using Laws' filters ( $3 \times 3$ ,  $5 \times 5$  and  $7 \times 7$ ).

correctly for the  $5 \times 5$  filter. The classification rates for each filter can be computed in a similar manner, and the rates are shown in Figure 7. As shown in Figure 7, filters  $5 \times 5$ ,  $7 \times 7$ ,  $7 \times 7_R$ ,  $9 \times 9$  and  $9 \times 9_R$  show fair classification rates.

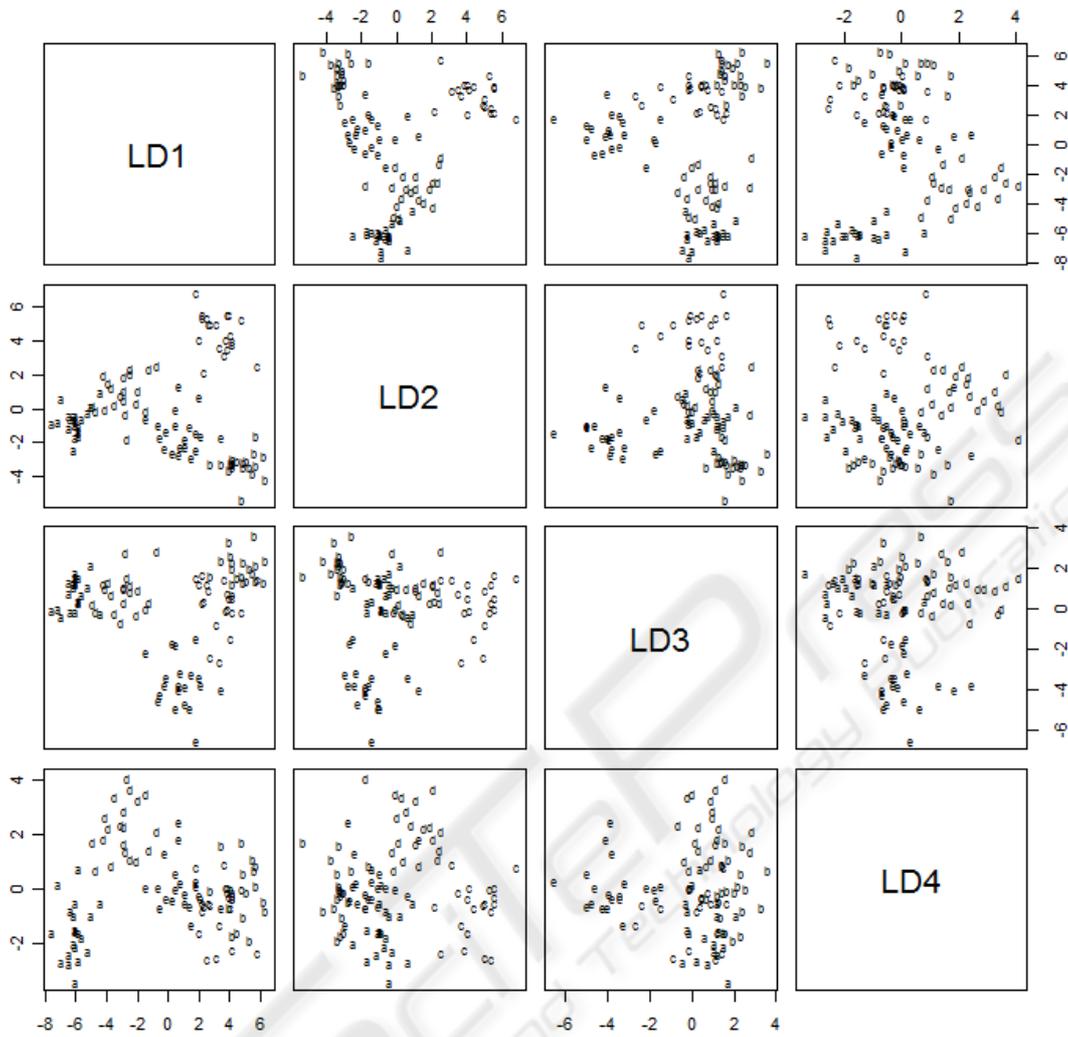


Figure 4: Texture data of discriminant coordinates for features of the  $7 \times 7$  filters.

LD1	LD2	LD3	LD4
0.5928	0.1757	0.1460	0.0855

Figure 5: Proportions of eigenvalues for discriminant functions ( $7 \times 7$  filters).

	a	b	c	d	e
a	9	0	0	1	0
b	0	9	0	0	1
c	0	0	10	0	0
d	0	0	1	9	0
e	0	0	0	1	9

Figure 6: Classification rate tables for filters  $5 \times 5$  (Seeds textures).

### 3.3 Similarity Retrievals of a Texture Database

Experiments on similarity retrievals of a texture database have been conducted. The Laws' filters

	$3 \times 3$	$5 \times 5$	$7 \times 7$	$9 \times 9$
Seeds	42.0%	92.0%	100.0%	96.0%
Paper	56.0%	90.0%	100.0%	96.0%
Pasta	56.0%	86.0%	100.0%	94.0%

	$3 \times 3_R$	$5 \times 5_R$	$7 \times 7_R$	$9 \times 9_R$
Seeds	42.0%	62.0%	78.0%	100.0%
Paper	38.0%	56.0%	90.0%	100.0%
Pasta	30.0%	64.0%	94.0%	100.0%

Figure 7: Classification rates for filters  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ ,  $9 \times 9$ ,  $3 \times 3_R$ ,  $5 \times 5_R$ ,  $7 \times 7_R$ ,  $9 \times 9_R$ .

( $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$  and  $9 \times 9$ ) have been applied to the database to extract texture image features. Also, rotation invariant features ( $3 \times 3_R$ ,  $5 \times 5_R$ ,  $7 \times 7_R$  and  $9 \times 9_R$ ) have been computed. The database contains 6380 textures ( $128 \times 128$ ; 20 images  $\times$  319 classes),

and the image feature extraction computation took about six hours using a standard computer (Intel (R) Core 2 Duo E8600 processor). The extracted features were analyzed by principal component analysis (PCA). The PCA transforms a number of correlated features into a smaller number of uncorrelated features called principal components. The PCA reduces the dimensionality of the data set without a significant loss of information. Since our image feature data sets contain redundant features for higher resolution filters such as  $7 \times 7$  and  $9 \times 9$ , the redundant features can be eliminated by applying the PCA. For our texture retrieval system, the dimensions of the feature space were reduced. All the dimensions of the feature space were reduced so that the features kept over 90% of their information.

The k-nearest neighbor algorithm (KNN) was used for retrieving texture from the database. A texture was classified by a majority vote of its neighbors, with the texture being assigned to the class most common amongst its  $k$  nearest neighbors.

Figure 8 shows examples of the similarity retrieval of textures. In each figure, the six most similar textures determined by the system are shown.

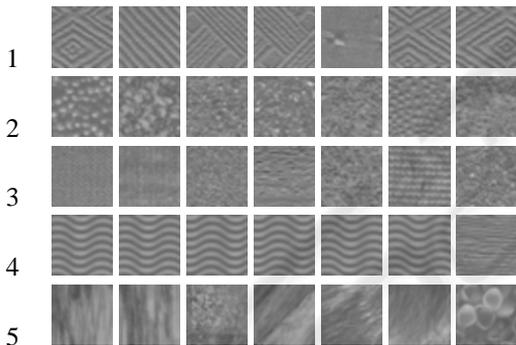


Figure 8: Similarity retrieval of textures (In each figure, the left image shows the query key).

Figure 9 shows a Recall-Precision graph for the retrieval test. In the experiment, 20 random query keys were selected, and corresponding recall rates and precision rates were computed.

In the figure, filters  $7 \times 7$ ,  $7 \times 7_R$ ,  $9 \times 9$  and  $9 \times 9_R$  show good retrieval results compared to other filters when our experimental database is examined.

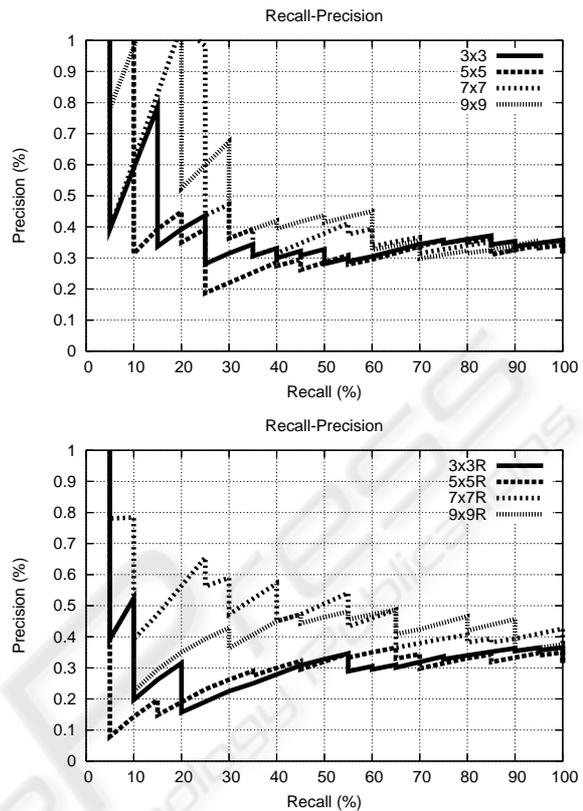


Figure 9: Recall-Precision graph for the retrieval test.

#### 4 CONCLUSIONS AND FUTURE WORK

Two dimensional image textures were analyzed by the Laws' texture energy measure approach. Various resolutions of Laws' filters ( $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$  and  $9 \times 9$ ) were used for extracting image features. Also, rotation invariant features ( $3 \times 3_R$ ,  $5 \times 5_R$ ,  $7 \times 7_R$  and  $9 \times 9_R$ ) were computed. A database of a texture image was analyzed using a simulation software program. In the experiments, classification rates of the Laws' filters were evaluated, and textures were classified fairly well when the filters were used in conjunction with the discriminant analysis. A principal component analysis (PCA) and a k-nearest neighbor (KNN) algorithm were used for the similarity retrieval. Use of these statistical techniques reduced unnecessary image features, and made possible the retrieval of pattern similar textures from the database.

For future experiments, additional 2D textures which contain various patterns will be examined. Other statistical approaches such as a quadratic discriminant analysis will be applied in conjunction with the Laws' filters for improving classification rates.

Also, other filtering techniques (Randen and Husoy, 1999) will be examined for content-based image retrieval.

## ACKNOWLEDGEMENTS

This research was partially supported by grants from the Telecommunications Advancement Foundation, Japan (TAF-2007) and the KAKENHI (YR-19700108).

## REFERENCES

- Chang, T. and Kuo, C. J. (1993). Texture analysis and classification with tree-structured wavelet transform. In *IEEE Trans. Image Process.* 2 (4), pp.429–441.
- Cross, G. and Jain, A. (1983). Markov random field texture models. In *IEEE Trans. Pattern Anal. Mach. Intell.* 5 (1), pp.25–39.
- Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. In *ACM Computing Surveys*, vol. 40, no. 2.
- Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. In *IEEE Trans. Systems Man Cybernetics*, Vol.3, No.6, pp.610–621. IEEE.
- Kato, T. (1992). Database architecture for content-based image retrieval. In *Proc. SPIE Image Storage Retrieval System*, 1662, pp.112–123.
- Laws, K. I. (1979). Texture energy measures. In *DARPA Image Understanding Workshop*, pp.47–51. DARPA.
- Laws, K. I. (1980). Rapid texture identification. In *Image Processing for Missile Guidance*, SPIE Vol. 238, pp.376–380.
- Manjunath, B. and Ma, W. (1996). Texture features for browsing and retrieval of image data. In *IEEE Trans. Pattern Anal. Mach. Intell.* 18 (8), pp.837–842.
- Ojala, T., Maenpaa, T., Pietikainen, M., Viertola, J., Kyllonen, J., and Huovinen, S. (2002). Outex - new framework for empirical evaluation of texture analysis algorithms. Proc. 16th International Conference on Pattern Recognition, Quebec, Canada, pp.701–706.
- Randen, T. and Husoy, J. (1999). Filtering for texture classification: a comparative study. In *Pattern Analysis and Machine Intelligence, IEEE Transactions on Vol. 21, Issue 4*, pp.291–310. IEEE.
- Suzuki, M. T. and Yaginuma, Y. (2007). A solid texture analysis based on three dimensional convolution kernels. In *Electronic Imaging 2007, Videometrics IX, (EI-2007), Proc. of SPIE and IST Electronic Imaging, SPIE Vol. 6491, 64910W* pp.1–8.
- Veltkamp, R. C. and Tanase, M. (2000). Content-based image retrieval systems: A survey. In *Technical Report UU-CS-2000-34*. Institute of Information and Computing Science, University Utrecht.