# The Effect of Sub-Sampling in Scale Space Texture Classification Using Combined Classifiers

M. J. Gangeh
Faculty of Engineering,
Multimedia University
Jalan Multimedia
63100 Cyberjaya,
Selangor, Malaysia

B. M. ter Haar Romeny
Biomedical Image analysis
Department of Biomedical
Engineering
Eindhoven University of
Technology
5600 MB Eindhoven
The Netherlands

C. Eswaran
Faculty of Information
Technology,
Multimedia University
Jalan Multimedia
63100 Cyberjaya,
Selangor, Malaysia

*Abstract*-**Textures show multi-scale properties and hence multiresolution techniques are considered appropriate for texture classification. Recently, the authors proposed a multiresolution texture classification system based on scale space theory and combined classifiers. However, the use of multiresolution techniques increases the computational load and memory space required. Sub-sampling can help to reduce these side effects of multiresolution techniques. However, it may degrade the overall performance of the classification system. In this paper the effect of sub-sampling is investigated in scale space texture classification using combined classifiers. It is shown that sub-sampling can help to reduce both computational load and memory space required without compromising the performance of the system.**

## I. INTRODUCTION

Multiresolution techniques become more and more important in texture classification due to the intrinsic multi-scale nature of textures [1-4]. Among the multiresolution techniques, scale space theory is a natural framework to construct multi-scale textures by deploying multi-scale derivatives up to certain order [5, 6]. The main issue in multiresolution techniques is the large feature space generated. In the literature, the most common approach for the construction of a feature space is the fusion of the features generated from different resolutions to come up with one feature space to be fed to a classifier. This generates a high dimensional feature space that may affect the performance of the classifier due to the 'curse of dimensionality' [7]. To tackle this problem typically severe feature reduction is applied in multiresolution techniques, e.g. by calculating moments of the histogram [5] or calculating the energy [8].

An alternative solution, based on combined classifiers, was recently proposed by the authors for this problem [9]. Using combined classifiers alleviates the problem of generating a large feature space, as the features generated from each scale/derivative are directly fed to an independent base classifier. In this approach, instead of concatenating features generated from each scale/derivative, the decisions made by the base classifiers are combined in a two-stage combined classifier.

In scale space texture classification, the patch sizes have to be increased at higher scales [9]. This is because at higher scales the coarser structures are emphasized and we look at these structures through larger windows. It has been shown in [9] that by increasing the patch sizes at higher scales, the performance of scale space texture classification system using combined classifiers is improved [9].

On the other hand, increasing the patch sizes at higher scales generates more features. As discussed in [10], by deploying principal component analysis (PCA) as feature extraction technique in our texture classification system, it can adaptively reduce the dimension of the feature space according to scale, i.e. more reduction is performed at higher scales.

In this paper the effect of sub-sampling at higher scales is to be investigated. It will be shown that based on the adaptive feature reduction property of PCA, sub-sampling does not affect the performance of the texture classification system. Better, it reduces the size of the datasets used for training and testing the system and this will reduce the size of memory required. It may also improve the performance of the system in terms of speed as the computation has to be performed on smaller datasets.

## II. THEORY

The texture classification system used in this paper is shown in Fig. 1. This structure was first introduced in [9] by the authors as a technique for multiresolution texture classification. As can be seen from Fig. 1, in this approach the features extracted from each scale are directly fed to a base classifier and instead of fusion of features obtained from different scales, the decisions made by these base classifiers are combined to decide on the class that a texture may belong to.

Here, the N-jet of derivatives up to the second order at multiple scales is used to construct the multi-scale textures. Then patches are extracted from these multi-scale textures to generate the feature space. As discussed in [9], the patch sizes are increased at higher scales to enable looking at coarser structures.

The method of feature reduction used in Fig. 1 is principal component analysis (PCA). In [10], it is shown that PCA can adaptively reduce the dimensionality of feature space according to the scale. This means that PCA reduces the feature space dimension more at higher scales due to less detailed structures obtained at higher scales.

## III. EXPERIMENTS

To compare the performance of the scale space texture classification using combined classifier (SSTCUCC) system with and without sub-sampling, two set of experiments are arranged. First, the performance of PCA is investigated in terms of the number of components required to maintain certain amount of cumulative fraction of variance with and without sub-sampling. Second, a supervised classification of some test images is performed using SSTCUCC with and without sub-sampling. The test images are from the well-known Brodatz album as shown in Fig. 2. Original Brodatz images are of size 640×640. However, in the experiments, the size is reduced to 512×512. The images have 256 gray levels.

### A. *Construction of Multi-Scale Texture Images*

The N-jet of all scaled derivatives up to the second order is chosen to construct the multi-scale texture images from each texture. Based on the steerability of Gaussian derivatives [9], we have used the zero$^{th}$ order derivative, i.e. the Gaussian kernel itself, $L_x$, $L_y$, $L_{xx}$, $L_{xy}$, $L_{yy}$, and $L_{xx} + L_{yy}$ where the last one is the Laplacian. For each derivative (including the zero$^{th}$ order) three scales are computed. The variances ($\sigma^2$) of the Gaussian derivatives in scales s1, s2, and s3 are 1, 4 and 7 respectively. In this way, out of each texture image, 7×3 texture images are obtained.

### B. *Preprocessing*

To make sure that for all textures the full dynamic range of the gray level is used contrast stretching is performed on all textures in different scales. Also, to make the textures indiscriminable to mean or variance of the gray level, DC cancellation and variance normalization are performed.

### C. *Construction of Train and Test Sets*

To ensure that the train and test sets are completely separate, they are extracted from the upper and lower half of each image respectively. For one texture, the patches have sizes of 18×18, 24×24, and 30×30 pixels before sub-sampling at scales s1, s2 and s3 respectively. After sub-sampling, the patch sizes are 18×18 at all scales. Test set size is fixed at 900. Train set size increases from 10 to 1500 to construct the learning curves.

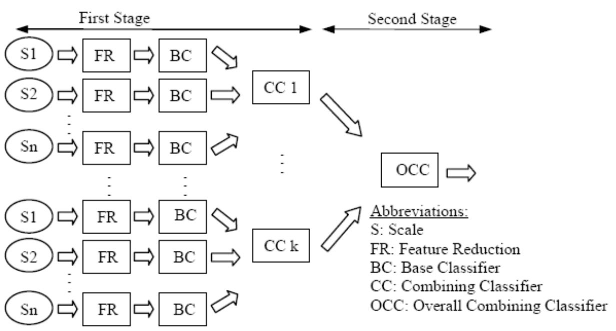



Figure 1. The structure of texture classification system used in this paper.

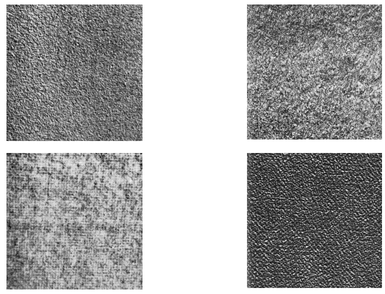


Figure 2. Textures D4, D9, D19 and D57 from Brodatz album used in the experiments.

### D. *Feature Extraction*

PCA is used for feature extraction in the combined classifier approach. The number of components used for dimension reduction is chosen to preserve 95% of the original variance in the transformed (reduced) space.

### E. *Classifier*

In the combined classifier approach, a two-stage parallel combined classifier is used as shown in Fig. 1. The type of base classifier used is a quadratic discriminant classifier (QDC); the type of combiner is fixed mean combiner for both stages as explained in [9]. Altogether, twenty one base classifiers, one for each scale in one derivative order are used. Since the number of components maintained in scale 1 is high even after applying PCA which may cause peaking phenomenon, regularization is used in QDC at scale 1 to resolve this problem.

### F. *Evaluation*

The performance of the texture classification system is evaluated by drawing the learning curves for training set sizes up to 1500. The error is measured 5 times for each training set size and the results are averaged.

## IV. RESULTS

Fig. 3 and Fig. 4 show the results of the first set of experiments, i.e. the performance of PCA with and without sub-sampling.

The effect of sub-sampling at scale 2 of the zero$^{th}$ order derivative is shown in Fig. 3. The top graph shows the cumulative fraction of variance (eigen-values) in respect to the number of components without sub-sampling; the bottom graph shows the same with sub-sampling. Only 100 components are shown to make the comparison easier; it must be reminded that since the patch size is 24×24 in scale 2, the total number of components before sub-sampling is 576 and after sub-sampling 324. As can be seen, only very small change can be noticed in the performance of PCA with and without sub-sampling.

Fig. 4 is similar to Fig. 3 but for scale 3 in the zero$^{th}$ order derivative. The feature space comprises 900 and 324 components before and after sub-sampling respectively. Only 50 components are shown in Fig. 4. As in the case of scale 2, the difference in the performance of PCA with and without sub-sampling is very small. Similar graphs are obtained for other derivative orders.

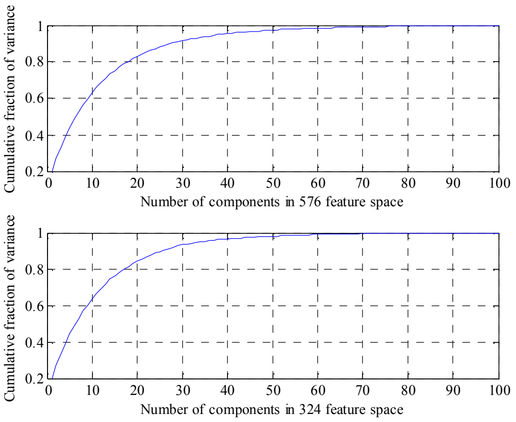



Figure 3. The cumulative fraction of variance (eigen-values) in respect to the number of components in scale 2 of the zero$^{th}$ order derivative before sub-sampling (*top graph*) and after sub-sampling (*bottom graph*).

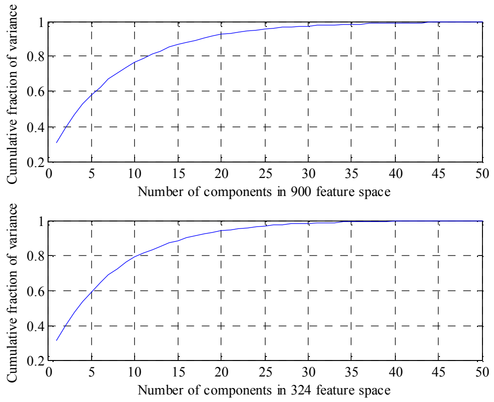



Figure 4. The cumulative fraction of variance (eigen-values) in respect to the number of components in scale 3 of the zero$^{th}$ order derivative before sub-sampling (*top graph*) and after sub-sampling (*bottom graph*).

Table I provides a comparison between the number of components required to retain 95% of the original variance after applying PCA with and without sub-sampling. Table II provides the same comparison to retain 99% of the original variance after applying PCA.

As can be seen from Table I, for PCA with 95% of retained variance, the maximum percentage of feature reduction using sub-sampling is 10.26% and 11.11% at scales 2 and 3 respectively. Similarly, from Table II, it can be noticed that for PCA with 99% of retained variance, it is 8.2% and 10.91% at scales 2 and 3 respectively. This means that sub-sampling may improve the feature reduction at scales 2 and 3 by only about 10%. However, it significantly reduces the size of the original datasets at scales 2 and 3 which is important in terms of memory space required for the calculations. This may also reduce the computational cost as the calculations have to be performed on smaller datasets.

Fig. 5 shows the results of scale space texture classification using two-stage combined classifiers with (top graph) and without (bottom graph) sub-sampling, i.e. the results of the second set of experiments as explained in the previous section. As can be seen, there is no significant difference in the performance of the classifier with and without sub-sampling.

TABLE I
THE PEROFORMACE OF PCA WITH 95% RETAINED FRACTION OF VARIANCE

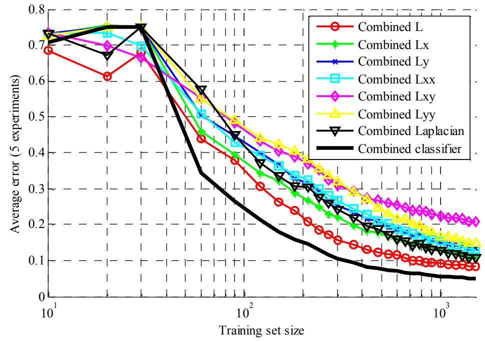
| Derivative-Scale | No. of components without sub-sampling | No. of components with sub-sampling | Percentage of feature reduction |
|---|---|---|---|
| $L$-$S_1$ | 142 | 142 | 0.00 |
| $L$-$S_2$ | 39 | 35 | 10.26 |
| $L$-$S_3$ | 24 | 22 | 8.33 |
| $L_x$-$S_1$ | 136 | 136 | 0.00 |
| $L_x$-$S_2$ | 40 | 37 | 7.50 |
| $L_x$-$S_3$ | 27 | 24 | 11.11 |
| $L_y$-$S_1$ | 138 | 138 | 0.00 |
| $L_y$-$S_2$ | 38 | 36 | 5.26 |
| $L_y$-$S_3$ | 25 | 24 | 4.00 |
| $L_{xx}$-$S_1$ | 185 | 185 | 0.00 |
| $L_{xx}$-$S_2$ | 46 | 42 | 8.70 |
| $L_{xx}$-$S_3$ | 29 | 27 | 6.90 |
| $L_{xy}$-$S_1$ | 162 | 162 | 0.00 |
| $L_{xy}$-$S_2$ | 56 | 51 | 8.93 |
| $L_{xy}$-$S_3$ | 34 | 31 | 8.82 |
| $L_{yy}$-$S_1$ | 195 | 195 | 0.00 |
| $L_{yy}$-$S_2$ | 43 | 40 | 6.98 |
| $L_{yy}$-$S_3$ | 27 | 25 | 7.41 |
| $L_{xx}+L_{yy}$-$S_1$ | 254 | 254 | 0.00 |
| $L_{xx}+L_{yy}$-$S_2$ | 63 | 58 | 7.94 |
| $L_{xx}+L_{yy}$-$S_3$ | 38 | 35 | 7.89 |

TABLE II
THE PERFORMANCE OF PCA WITH 99% RETAINED FRACTION OF VARIANCE

| Derivative-Scale | No. of components without sub-sampling | No. of components with sub-sampling | Percentage of feature reduction |
|---|---|---|---|
| $L$-$S_1$ | 246 | 246 | 0.00 |
| $L$-$S_2$ | 66 | 61 | 7.58 |
| $L$-$S_3$ | 38 | 35 | 7.89 |
| $L_x$-$S_1$ | 215 | 215 | 0.00 |
| $L_x$-$S_2$ | 63 | 58 | 7.94 |
| $L_x$-$S_3$ | 41 | 37 | 9.76 |
| $L_y$-$S_1$ | 219 | 219 | 0.00 |
| $L_y$-$S_2$ | 61 | 56 | 8.20 |
| $L_y$-$S_3$ | 39 | 36 | 7.69 |
| $L_{xx}$-$S_1$ | 257 | 257 | 0.00 |
| $L_{xx}$-$S_2$ | 72 | 67 | 6.94 |
| $L_{xx}$-$S_3$ | 45 | 41 | 8.89 |
| $L_{xy}$-$S_1$ | 223 | 223 | 0.00 |
| $L_{xy}$-$S_2$ | 83 | 77 | 7.23 |
| $L_{xy}$-$S_3$ | 51 | 46 | 9.80 |
| $L_{yy}$-$S_1$ | 267 | 267 | 0.00 |
| $L_{yy}$-$S_2$ | 69 | 64 | 7.25 |
| $L_{yy}$-$S_3$ | 43 | 39 | 9.30 |
| $L_{xx}+L_{yy}$-$S_1$ | 304 | 304 | 0.00 |
| $L_{xx}+L_{yy}$-$S_2$ | 90 | 83 | 7.78 |
| $L_{xx}+L_{yy}$-$S_3$ | 55 | 49 | 10.91 |

## V. CONCLUSION

We conclude that sub-sampling can save the memory space for the computations and improve the computation speed. However, it almost has no effect on the performance of the combined classifier in scale space texture classification. This is mainly due to the adaptive feature reduction property of PCA at multiple scales.



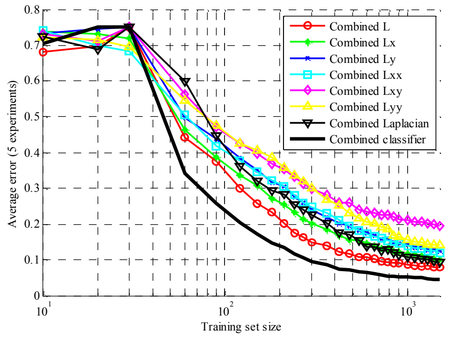




Figure 5. The results of scale space texture classification using combined classifiers with (*top graph*) and without (*bottom graph*) sub-sampling.

## APPENDIX

In these experiments matrices *Sa* and *Sb* given below are used for the sub-sampling of patches of 24×24 and 30×30 to 18×18 respectively. These matrices are not unique and there are other possibilities. However, it will not affect the overall performance of the sub-sampling.

$$
Sa = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}
$$

$patch_S = Sa \cdot pacth \cdot Sa'$ where $patch_S$ is a sub-sampled patch and the operator used is the inner product.

$$
Sb = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}
$$

Similarly, a sub-sampled patch at scale 3 can be obtained using: $patch_S = Sb \cdot patch \cdot Sb'$ .

## REFERENCES


[1] E. Hadjidemetriou, M. D. Grossberg, and S.K. Nayar, "Multiresolution histograms and their use for recognition", *IEEE Trans. on PAMI*, vol. 26, pp. 831-847, Jul. 2004.

[2] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", *IEEE Trans. on PAMI*, vol. 24, pp. 971-987, Aug. 2002.

[3] L. Wang and J. Liu, "Texture classification using multiresolution Markov random field models", *Pattern Recognition Letters*, vol. 20, pp. 171-182, Feb. 1999.

[4] S. T. Li and J. Shawe-Taylor, "Comparison and fusion of multiresolution features for texture classification", *Pattern Recognition Letters*, vol. 26, pp. 633-638, Apr. 2005.

[5] B. van Ginneken and B. M. ter Haar Romeny, "Multi-scale texture classification from generalized locally orderless images", *Pattern Recognition*, vol. 36, pp. 899-911, Apr. 2003.

[6] I. C. Sluimer, P. F. van Waes, M. A. Viergever, and B. van Ginneken, "Computer-aided diagnosis in high resolution ct of the lungs", *Medical Physics*, vol. 30, pp. 3081-3090, Dec. 2003.

[7] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: a review", *IEEE Trans. on PAMI*, vol. 22, pp. 4-37, Jan. 2000.

[8] T. Randen and J. H. Husoy, "Filtering for texture classification: a comparative study", *IEEE Trans. on PAMI*, vol. 21, pp. 291-310, Apr. 1999.

[9] M. J. Gangeh, B. M. ter Haar Romeny, and C. Eswaran, "Scale space texture classification using combined classifiers", In: B. K. Ersboll, and K. S. Pedersen (eds.), 15th Scandinavian Conference on Image Analysis, LNCS, vol. 4522, pp. 324-333, 2007.

[10] M. J. Gangeh, R. P. W. Duin, C. Eswaran, and B. M. ter Haar Romeny, "Scale space texture classification using combined classifiers with application to ultrasound tissue characterization", In: F. Ibrahim, N.A. Abu Osman, J. Usman, and N.A. Kadri (eds.), IFMBE Proceedings, vol. 15, pp. 303-306, 2006.