

Multi-Scale Binary Patterns for Texture Analysis

Topi Mäenpää and Matti Pietikäinen

Machine Vision Group, Infotech Oulu
University of Oulu, Finland

Abstract. This paper presents two novel ways of extending the local binary pattern (LBP) texture analysis operator to multiple scales. First, large-scale texture patterns are detected by combining exponentially growing circular neighborhoods with Gaussian low-pass filtering. Second, cellular automata are proposed as a way of compactly encoding arbitrarily large circular neighborhoods. The performance of the extensions is evaluated in classifying natural textures from the Outex database.

1 Introduction

The LBP operator, introduced by Ojala *et al.* has been shown to be a powerful measure of image texture [1]. In its original form, the operator works by thresholding a 3×3 neighborhood with the value of the center pixel, thus forming a local binary pattern, which is interpreted as a binary number. The occurrences of different local patterns are collected into a histogram that is used as a texture descriptor.

The most prominent limitation of the LBP operator has been its small spatial support area. Features calculated in a local 3×3 neighborhood cannot capture large-scale structures that may be the dominant features of some textures. The operator is not very robust against local changes in the texture, caused for example by varying viewpoints or illumination directions. Therefore, an operator with a larger spatial support area is needed.

In a recent paper, the operator was extended to facilitate rotation invariant analysis of image textures at multiple scales [2]. In the general definition of the operator, arbitrary circular neighbor sets are used instead of the eight-neighbors. The number of samples as well as the sampling radius can vary. In addition, operators with different parameters can be combined to obtain a multi-scale description of texture. In Fig. 1, three neighborhoods with a varying number of samples (P) and different neighborhood radii (R) are shown. The corresponding LBP operators are denoted by $LBP_{P,R}$. Samples that do not exactly fall on pixels are obtained with bilinear interpolation. The value of the center pixel (gray) is used as a threshold in producing a P -bit binary code that describes the local pattern in the texture.

The number of bins in the LBP distributions can be reduced by considering only “uniform” LBP codes. These are local patterns in which at most two zero-to-one or one-to-zero transitions are allowed in the circular presentation of the

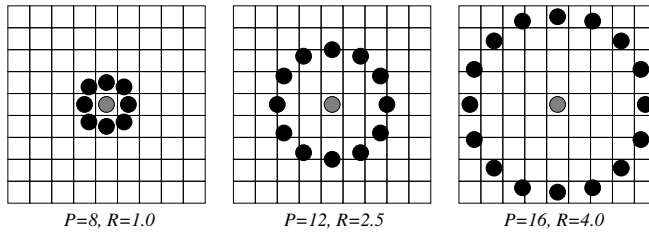


Fig. 1. Circularly symmetric neighbor sets

binary number. These codes have been shown to dominate the LBP distribution. The resulting operators are denoted by $LBP_{P,R}^{u_2}$.

In [2], a multi-resolution LBP was constructed by extracting a number of LBP codes for each pixel with different P and R values. The marginal distributions of these codes were used as a texture descriptor. This approach has some shortcomings, as detailed in the following.

From a signal processing point of view, the sparse sampling exploited by LBP operators with large neighborhood radii may not result in an adequate representation of the two-dimensional image signal. Aliasing effects are an obvious problem. So might be noise sensitivity as sampling is made at single pixel positions, without low-pass filtering. One might argue that collecting information from a larger area would thus make the operator more robust. From the statistical point of view, even sparse sampling is however acceptable provided that the number of samples is large enough. To solve this problem, an exponentially growing multi-resolution LBP combined with Gaussian filtering is introduced in Sect. 2.

Even with low-pass filtering, the number of different parameter combinations cannot be made very large. Since each parameter combination adds a whole new distribution to the feature vector, classification soon becomes impossible. In this paper, cellular automata are proposed as a possible way of avoiding this. Section 3 describes how cellular automaton rules can be deduced from multiple LBP codes and how the rule codes can be used in compactly encoding arbitrarily large neighborhoods. To our knowledge, this is the first time cellular automata have been used in extracting texture features for classification.

2 Gaussian Low-Pass Filtering

The LBP operator can be combined with multi-scale filtering in a straightforward way. Using Gaussian low-pass filters, each sample in the neighborhood can be made to collect intensity information from an area larger than the original single pixel. The filters and sampling positions are designed to cover the neighborhood as well as possible while minimizing the amount of redundant information. As a consequence, the radii of the LBP operators used in the multi-resolution version grow exponentially.

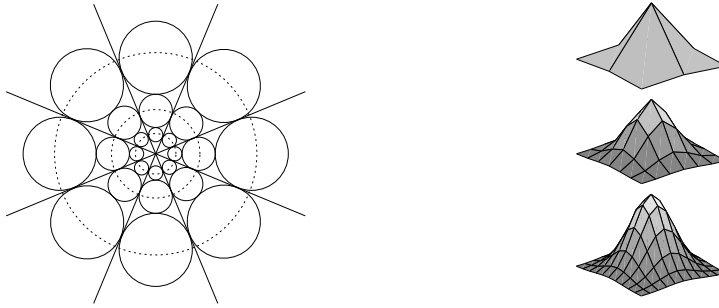


Fig. 2. The effective areas of filtered pixel samples in an eight-bit multi-resolution LBP operator (left), and Gaussian low-pass filters for scales 2, 3, and 4 (right)

Fig. 2 shows a neighborhood quantized angularly to eight sectors. Without filtering, an LBP code would be constructed by sampling the neighborhood at the centers of the solid circles. With large radii, the distance between samples is large, presumably making the codes unreliable. With a low-pass filter, the intensity information for a sample is collected from a larger area, indicated by the solid circles. The outer radius of this “effective area” with respect to the center of the neighborhood is given by

$$r_n = r_{n-1} \left(\frac{2}{1 - \sin(\pi/P_n)} - 1 \right), \quad n = 2, \dots, N, \quad (1)$$

where N is the number of scales, and P_n is the number of neighborhood samples at scale n . Since low-pass filtering is useful only with radii larger than one for $P_1 = 8$, r_1 is set to 1.5, which is the shortest distance between the center and the border of a 3×3 neighborhood.

The radii of the LBP operators are chosen so that the effective areas touch each other. Consequently, the radius for an LBP operator at scale n ($n \geq 2$) is halfway between r_n and r_{n-1} :

$$R_n = \frac{r_n + r_{n-1}}{2}. \quad (2)$$

These radii are illustrated with dotted circles in the figure.

The effective areas are realized with Gaussian low-pass filters designed so that 95% of their mass lies within the solid circles. The effective areas could be designed so that they filled the sampling sector more precisely by utilizing filters that are not circularly symmetrical. But then a different filter should be used for each orientation. Circularly symmetrical filters have the advantage that only one filter is needed for each scale.

The procedure used in building a multi-resolution filtered LBP is very similar to that used by Ojala *et al.* [2]. The only difference is in that neighborhood samples with radii greater than one are obtained via low-pass filtering. Furthermore, neighborhood radii are not chosen arbitrarily, but so that the effective areas touch each other. We will call the operator LBP (F for filtering).

3 Cellular Automata

Cellular automata can be generally described as discrete dynamical systems completely defined by a set of rules in a local neighborhood. The state of a system is represented as a regular grid, on which the rules are applied to produce a new state. An interesting property of cellular automata is that very simple rules can result in very complex behavior. A few people have used cellular automata in image processing (see e.g. [3, 4]). It however seems that the use of cellular automata in extracting features for image analysis has been largely overlooked.

In encoding binary neighborhoods, we used one-dimensional cellular automata in which the value of a sample at scale n is determined by its three closest neighbors at scale $n - 1$, resulting in 256 different cellular automaton rules. Fig. 3 (a) displays an example of an automaton and a pattern it has produced. The automaton number of this particular automaton is $10001001_2 = 137_{10}$. In Fig. 3 (b), two $LBP_{8,R}$ codes are dressed to form the two topmost rows of a two-dimensional pattern. Since the rows are treated circularly, the breakpoint (dashed line) plays no role. This property has an important consequence: the cellular automata are always invariant with respect to rotation, irrespective of the LBP version used as the input signal.

Any number of LBP scales can be added to the pattern. The problem of encoding a neighborhood now becomes finding a cellular automaton rule that could have produced the pattern, starting from the input signal at the topmost row. Arbitrarily large neighborhoods can thus be encoded with the input signal (P bits) and a cellular automaton rule (eight bits). The joint distribution of the input signal (the LBP code at scale 1) and the cellular automaton rule code can be used as a texture descriptor.

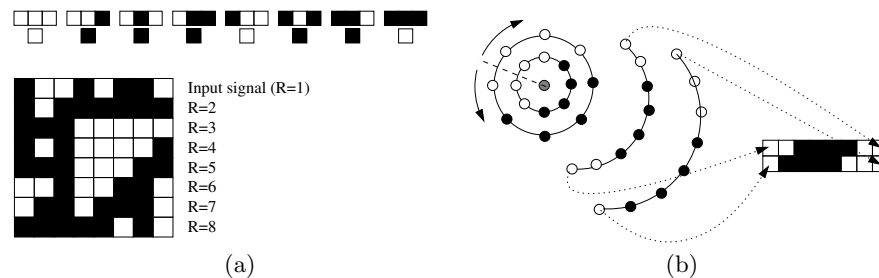


Fig. 3. A cellular automaton rule and a pattern it has produced (a). Turning a multi-resolution LBP into a two-dimensional pattern (b).

A straightforward way of finding the most appropriate cellular automaton rule is to try out every possibility, and to compare the result to the observed pattern. However, searching through all possible rules is a time consuming operation. To avoid exhaustive searching probabilistic cellular automata were used instead. In a probabilistic automaton, the decision whether the value of a sam-

ple at scale n will be one or zero is based on the neighborhood scale $n - 1$ and two probabilities assigned to the possible outcomes. If both probabilities are 0.5, then a certain neighborhood is equally likely to produce either one or zero. The problem is that the automaton rule must now be encoded with eight real numbers instead of eight bits.

An obvious way of converting probabilistic rules to conventional ones is to binarize the probabilities. This way, the larger probability always “wins”, and the rule can again be represented with eight bits. This simple method has the drawback that with small patterns, not enough information can be collected to draw statistically sound decisions. Therefore, the number of scales in the multi-scale LBP operator should be large. Due to border effects, a large neighborhood means that a large texture must be used. Since the correlation between the center and a neighbor decreases with distance, very large neighborhoods may not provide much useful information. With small textures, only unreliable measurements can be made, which may render the joint distribution of LBP codes and cellular automaton rules both too sparse and too noisy.

To fight the imminent problems with statistical stability, the number of “accepted” cellular automaton rules can be reduced. Based on knowledge of the structure of digital images, one may argue that certain rules are more likely to occur than others. For example, due to the high correlation between neighboring pixels, it is likely that a sequence of three “ones” at scale 1 is likely to produce a “one” at scale 2. An easy way of purging unlikely cellular automata is to use only the most common ones. With a large amount of natural textures, reliable statistics of the occurrences of different cellular automata can be derived. Then, the number of applicable automata can be reduced by discarding the least frequently occurring ones. Even more compact descriptor can be achieved by considering only the marginal distributions of the LBP codes and cellular automaton rules. This method has also been applied to the multi-resolution LBP [2]. As it turns out, the latter approach works much better.

As a texture feature, the marginal distributions of $LBP_{P,1}$ codes and cellular automata rules deduced from S consecutive scales is used, and denoted by $LBPCA_{P,S}$. For each pixel, S LBP codes are calculated, and the automaton rule is deduced from these codes. The distribution of rules is used as a multi-scale texture descriptor, and it is concatenated to the $LBP_{P,1}$ feature vector.

4 Experimental Results

As a test bed for the proposed extensions, a classification experiment with 24 natural textures from the Outex database [5] was arranged. The textures are shown in Fig. 4. A prebuilt test set (Outex_TC_00011) was used for this purpose. In the test suite, twenty samples of each texture class are created for training by extracting non-overlapping 128×128 sub-images of the source texture, imaged at a 100dpi spatial resolution. As testing data, twenty samples of each of the textures imaged at a 120dpi resolution are used. Thus, there are 480 texture samples for both training and testing. Some examples of the scale difference are

shown in Fig. 5. The suite allows one to inspect the robustness of a texture measure against relatively small (20%) changes in the spatial scale of textures. As a classification principle, the 3-NN method was used with the log-likelihood measure proposed for the LBP by Ojala *et al.*[2]. The results are summarized in Tbl. 1.

Table 1. Classification results

Method	Bins	Score
$\text{LBP}_{8,1}$	256	92.7
$\text{LBP}_{8,1}^{u^2} +_{16,2}^{u^2} +_{24,3}^{u^2}$	857	96.3
$\text{LBP}_{8,1}^{u^2} +_{16,3}^{u^2} +_{24,5}^{u^2}$	857	96.3
$\text{LBP}_{8,1}^{u^2} +_{8,2.4}^{u^2} +_{8,5.4}^{u^2}$	177	98.1
$\text{LBP}_{8,1}^{u^2} +_{8,2.4}^{u^2} +_{16,4.2}^{u^2} +_{16,6.2}^{u^2}$	604	99.2
$\text{LBPF}_{8,1}^{u^2} +_{8,2.4}^{u^2} +_{8,5.4}^{u^2}$	177	99.0
$\text{LBPF}_{8,1}^{u^2} +_{8,2.4}^{u^2} +_{16,4.2}^{u^2} +_{16,6.2}^{u^2}$	604	99.6
$\text{LBPCA}_{8,12}$	512	99.4
$\text{LBPCA}_{8,12}^{u^2}$	315	99.4

For the $\text{LBP}_{8,1}$, the test suite is a hard problem, although most of the texture classes are classified with a 100% accuracy. Misclassified samples in five texture classes drop the final accuracy to 92.7%. The three-scale $\text{LBP}_{8,1}^{u^2} +_{16,2}^{u^2} +_{24,3}^{u^2}$ operator that has previously shown very good performance is clearly better with a 96.3% score. The same result was also obtained with $\text{LBP}_{8,1}^{u^2} +_{16,3}^{u^2} +_{24,5}^{u^2}$.

With a three-scale $\text{LBP}_{8,R}^{u^2}$ accompanied with Gaussian low-pass filtering an accuracy of 99.0% was obtained. At the first sight, this seems a reasonable performance enhancement compared to the “old” multi-resolution versions. Most of the difference can however be attributed to the exponentially growing neighborhood size; the score of the same operator without low-pass filtering was 98.1%. The filtering does however seem to be somewhat helpful, as the result without it is consistently slightly worse.

The joint distribution of LBP codes and cellular automaton rules proved to be too sparse to be statistically reliable, even when infrequently occurring entries were removed. It turned out that on average, 96% of the mass of the joint distribution could be covered by considering only 6% of the bins. Even then, the resulting distributions contained over 4000 bins, resulting in statistical unreliability, very slow classification and excessive memory consumption. Furthermore, the accuracy was disappointing.

Statistical reliability seems to be the key issue in using the distributions of cellular automaton rules. The concatenated marginal distributions of LBP codes and cellular automaton rules turned out to be a very powerful measure of image texture. Consistent with the hypotheses, the accuracy of the method increased with the number of LBP scales in use, until a certain limit. The size of the resulting descriptors does however remain constant. The best score, 99.4%, was obtained with a twelve-scale $\text{LBPCA}_{8,12}$. The same score was achieved by a version with only “uniform” LBP codes enabled.

5 Discussion and Conclusions

The multi-resolution LBP has been shown to be a powerful measure of image texture [2,6]. Until now, its main limitations have been sparse sampling and an inability to cope with a large number of different local neighborhoods. In this paper, two techniques were presented as possible solutions to these problems. Gaussian low-pass filters are used in collecting texture information for the multi-scale LBP not only from a single pixel but a carefully designed “effective area”. The increase in classification accuracy may however not be large enough, compared to the increased computational burden. Furthermore, the size of the LBP distribution cannot be made shorter with filtering.

A novel way of encoding arbitrarily large binary neighborhoods with cellular automata was presented. The method was used in compactly encoding even 12-scale LBP operators. A feature vector containing the marginal distributions of LBP codes and cellular automaton rules turned out to be an excellent multi-resolution texture descriptor. Nevertheless, the results need still to be confirmed with a larger amount of data and with different setups. Furthermore, it is likely that the number of cellular automaton rules used for texture description can be reduced to obtain an even more compact descriptor.

Acknowledgements

The financial support provided by the Academy of Finland, the national Graduate School in Electronics, Telecommunication, and Automation, and Nokia Foundation is gratefully acknowledged.

References

1. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition* **29** (1996) 51–59
2. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray scale and rotation invariant texture analysis with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (2002) 971–987
3. Hernandez, G., Herrann, H.: Cellular automata for elementary image enhancement. *Graphical Models and Image Processing* **58** (1996) 82–89
4. Ikenaga, T., Ogura, T.: Real-time morphology processing using highly parallel 2-d cellular automata CAM₂. *IEEE Transactions on Image Processing* **9** (2000) 2018–2026
5. Ojala, T., Mäenpää, T., Pietikäinen, M., Viertola, J., Kyllönen, J., Huovinen, S.: Outex - new framework for empirical evaluation of texture analysis algorithms. In: 16th International Conference on Pattern Recognition. Volume 1., Québec, Canada (2002) 701–706 <http://www.outex.oulu.fi/>.
6. Mäenpää, T., Pietikäinen, M., Ojala, T.: Texture classification by multipredicate local binary pattern operators. In: 15th International Conference on Pattern Recognition. Volume 3., Barcelona, Spain (2000) 951–954

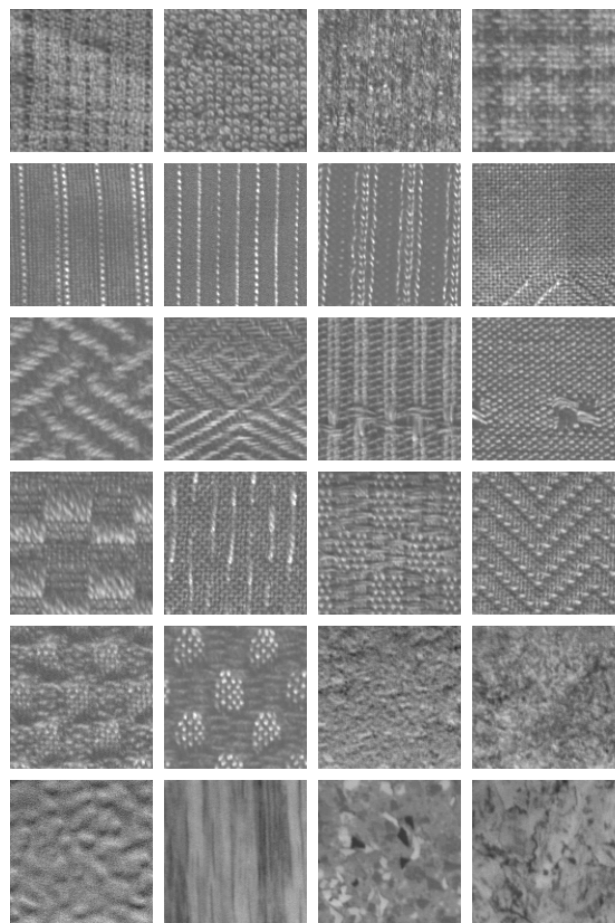


Fig. 4. The 24 Outex textures used in experiments

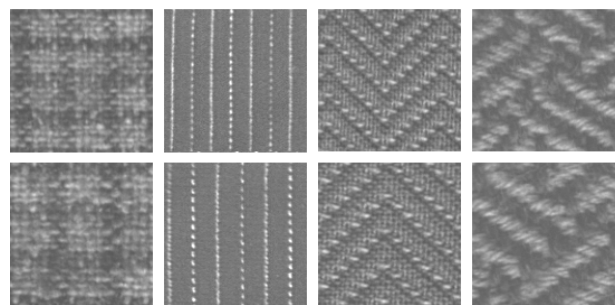


Fig. 5. Examples of the scale difference between training and testing samples. The top row contains training samples (100dpi), and the bottom row the corresponding testing samples (120dpi).