Analysing Micro- and Macro-Structures in Textures

B. M. Waller, M. S. Nixon, J. N. Carter
School of Electronics and Computer Science
University of Southampton
Southampton, UK
{bmw306,msn,jnc}@ecs.soton.ac.uk

***Abstract*—Analysing micro- and macro-structures within images confers ability to include scale in texture analysis. Filtering allows for selection of texture structures at different scales, revealing the micro- and macro-structures which would otherwise be concealed. The new approach to texture segmentation uses low- and high-pass filters to achieve this scale-based analysis. Segmentation is performed using Local Binary Patterns as an example of the type of feature vector that can be used with the new process. These are generated for the original image and each of the filtered images. A two stage training process is used to learn the optimum filter sizes and to produce model histograms for each known texture class. These are used in the supervised segmentation of texture mosaics generated from the VisTex database. The results demonstrate the superiority of the new combined approach compared to the best multi-resolution LBP configuration and analysis only using lowpass filters. Noise analysis has also confirmed the advantageous properties of low- and high-pass filtering, and confirms that it is optimal to combine the two forms in texture segmentation.**

***Keywords*-texture; LBP; filtering; highpass; lowpass**

I. Introduction

Natural images contain many different textures at different scales. Also, depending on camera viewpoint, instances of one texture may be present at different scales. Even samples of texture in databases such as VisTex [6] do not contain a single texture at a single scale. Applications can be optimised to capture a specific texture within the image, but this disregards information that could improve the segmentation or classification rates for that sample.

The large scale components of an image are known as macro-structures and the small scale components are known as micro-structures. Image filtering can be used to remove the structures from certain scales of the image. Micro-structures, which are those that repeat the most throughout the image, tend to be present in the high frequency components of the image. Lowpass filtering therefore can be used to remove these high frequencies, and hence micro-structures, from the image. Similarly, highpass filtering can be used to remove the macro-structures from the image. As the cutoff frequency of the filter is changed to remove more frequencies, more structures will be removed from the image. It is possible therefore, to analyse images in the absence of certain texture scales, thereby enhancing the effect of the remaining components on segmentation performance.

Previous work on using filtering to construct a multi-scale texture descriptor includes a paper by Turtinen and Pietikäinen [8], where a multi-scale feature vector for each pixel was extracted by taking three squares of increasing size around the pixel and resizing the larger two to the dimensions of the smallest using Gaussian filtering and downsampling. A Local Binary Pattern (LBP) histogram was computed for each square and the histograms concatenated together into a single feature. He *et al.* [1] used a Gaussian pyramid to obtain features at different scales and concatenated the histograms in a similar manner. Both papers focussed on macro-structures, with micro-structures only obtained from the original unfiltered image in each case. Since filtering the image with a lowpass Gaussian filter exposes the macro-structures we can use a highpass filter to expose the micro-structures.

In this paper we introduce a technique of expanding existing texture analysis operators to take advantage of the composition of structures at different scales. Traditionally, a feature vector is obtained from the sample image which is compared to feature vectors from training images for classification or segmentation. In this process, called Accumulative Filtering, the sample image is filtered separately with a number of different highpass and lowpass filters with varying cutoff frequencies. This produces a set of images, each with a different range of frequencies (and hence texture structures) removed. Feature vectors are obtained from each of the filtered images and concatenated together into a single multi-scale feature vector. This is then used for the classification or segmentation of the original unfiltered image. We offer Local Binary Patterns (LBP) as an example of a texture analysis operator that can be extended using our Accumulative Filtering process, however any other operator which provides a feature vector could equally be considered. Tests using LBP histograms as the feature vector demonstrate a greatly improved segmentation accuracy across the entire image, including texture boundaries.

II. Multi-scale Local Binary Patterns

The basic Local Binary Pattern (LBP) operator [4] provides a code representing the texture pattern for a pixel based on a 3x3 pixel area of the image and is considered too small for images containing larger scales. Mäenpää *et al.* [2] introduced a multi-predicate LBP which increased the area from which the LBP code is calculated. The histograms from various predicates are concatenated together to form a single multi-scale description of the texture.

This was found to provide improved results over those obtained from the basic LBP. Ojala *et al.* [5] extended this further with the multi-resolution rotation invariant LBP which calculated the LBP code from *P* points on a circle of radius *R*. This method of increasing the size of the LBP operator enables it to capture the larger scales in the image which would otherwise be missed, but it must still be combined with the basic LBP to ensure that the smaller scale elements of the image are also captured. This has the same effect as the process of lowpass filtering followed by downsampling seen in [8] and [1]; instead of increasing the size of LBP, the image is reduced in size.

The rotation invariant uniform LBP code for a point, $LBP_{P,R}^{riu2}$ [5], is calculated by:

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{if } U(LBP_{P,R}) \leq 2 \\ P+1 & \text{otherwise} \end{cases} \tag{1}$$

where the number of bit changes is

$$\begin{aligned} U(LBP_{P,R}) &= |s(g_{P-1} - g_c) - s(g_0 - g_c)| \\ &+ \textstyle\sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \end{aligned} \tag{2}$$

and $s(x)$ is 1 if $x \geq 0$ and 0 otherwise. g_0 to g_{P-1} are the grey values of the interpolated points on the circle around the central pixel, grey value g_c, for which the LBP code is calculated.

III. Accumulative Filtering

In conventional segmentation, a texture class is assigned to a pixel dependant on a training database and a chosen distance metric. Although accuracy can appear high due to spatial coverage, detail can be lost as in Figure 1(b). For mislabelled pixels the correct texture class is often at a close distance to the selected class. To compensate for this the image can be processed by a texture operator of different size, or can be filtered to enhance cohesion of texture classes.

Our new Accumulative Filtering (AF) algorithm segments an image by filtering the image multiple times each with a different cutoff frequency and then segmenting the original image and each filtered image concurrently. A final decision is made on the texture class for each pixel by combining the results from each filtered image using the log likelihood measure in Equation 5, which has the effect of adding the distances obtained for each texture class from each filtered image. The larger the number of filtered images included in this process, the higher the segmentation accuracy. In Figure 1, the original image in 1(a) is filtered with a lowpass and highpass filter to produce the images in 1(c) and 1(d). The result in 1(b) is obtained from segmentation using the feature vectors obtained from the original image. In 1(e), the feature vectors from 1(a) and 1(c) are combined and the result in 1(f) is from the combination of all three images. The segmentation accuracies show that each time another feature vector is added the result improves. This can also be determined from a visual inspection of the noise within each segment of the results. Segmentation of the filtered images on their own gives an increasingly poor result as more frequencies are removed, shown in Figure 2, however Figure 3 shows how the combination of the feature vector from one filtered image with the original feature vector has a positive effect on the segmentation accuracy. For both highpass and lowpass filtering, all filter sizes within a range increase the segmentation accuracy when compared with the original segmentation shown by a green horizontal line on the graphs.

A. Filtering Process

Images are filtered in the frequency domain as per Equations 3 and 4 which describe lowpass and highpass filtering.

$$\mathcal{F}_{LP}(u,v) = \begin{cases} \mathcal{F}(u,v) & \text{if } \sqrt{u^2+v^2} \leq f \cdot w \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$\mathcal{F}_{HP}(u,v) = \begin{cases} 0 & \text{if } \sqrt{u^2+v^2} \leq f \cdot w \\ \mathcal{F}(u,v) & \text{otherwise} \end{cases} \tag{4}$$

The width of the image in pixels, w, is necessary for the filtering process because larger images are represented by a larger range of spatial frequencies in the frequency domain. This is important during supervised image segmentation when sample and model images are of different sizes. The filter size, f, is related to the cutoff frequency of the filter. Lowpass filtered images could be downsampled, but image size is retained to allow later comparison with highpass filtered images. More sophisticated filter mechanisms could have been used, but our premise here is to explore whether frequency domain filtering (and particularly highpass) can be used to explore scale to advantage in texture segmentation.

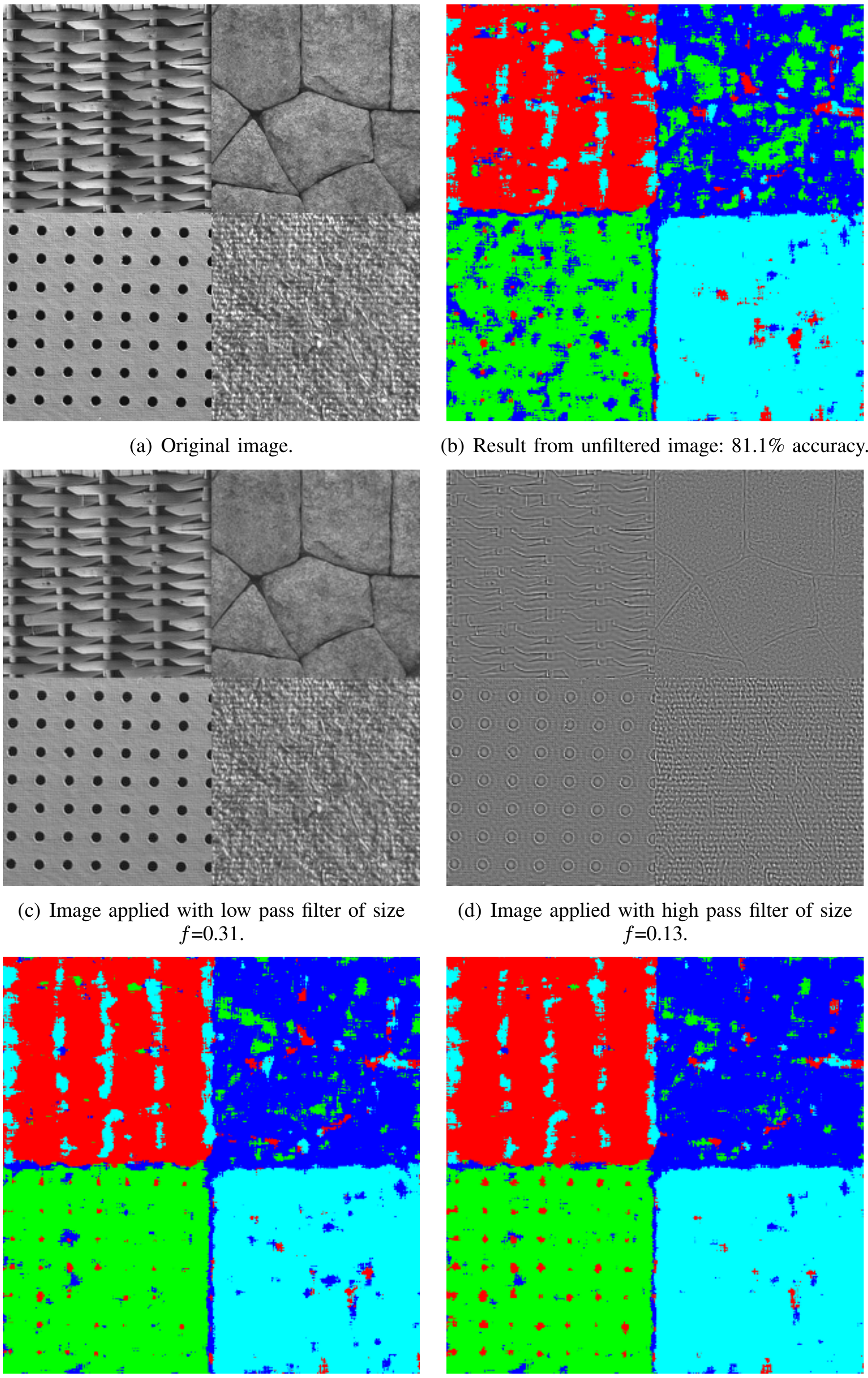
B. Segmentation Algorithm

The Accumulative Filtering process requires a texture descriptor and comparison algorithm as a preprocessor to provide a distance between a sample and each known texture class for each filtered image. Due to their popularity and potency, we have chosen Uniform Local Binary Patterns (LBP) [5] as our texture descriptor and Histogram Comparison (HC) [2] is used as the distance metric. This choice is not fundamental to the success of our process and any other alternative approach which can provides these distances would be equally applicable.

The uniform LBP is described in Equations 1 and 2 and the output from the HC algorithm is a distance from each pixel in the image to each of the possible texture classes. Our AF process provides multiple histograms for each texture class; one for each filter size used. A two-dimensional log likelihood metric is used [2] to calculate the distance between sample and model in this case:

$$L(S,M) = -\sum_{f \in A} \sum_{n=1}^{N} S_{fn} \ln M_{fn} \tag{5}$$

where N is the number of histogram bins, S_{fn} and M_{fn} are the probabilities of bin *n* in histogram f for the sample and model respectively and A is the set of filter sizes



(a) Original image.

(b) Result from unfiltered image: 81.1% accuracy.

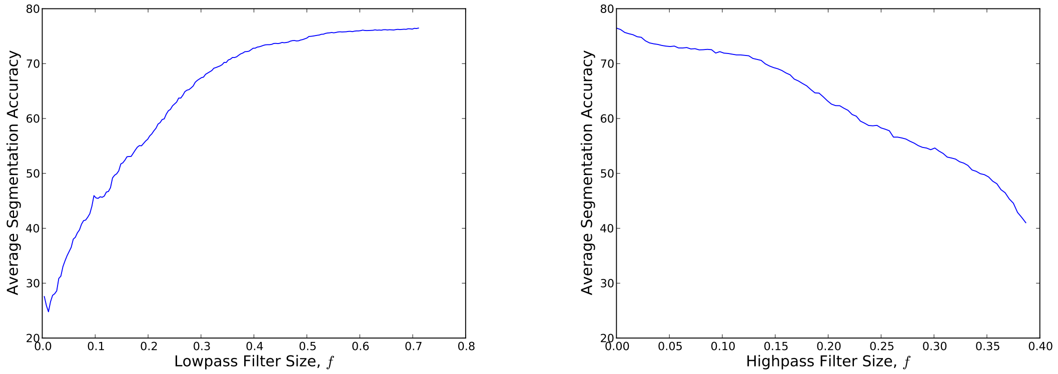
(c) Image applied with low pass filter of size f=0.31.

(d) Image applied with high pass filter of size f=0.13.

(e) Result from lowpass f=0.31 combined with unfiltered: 88.5% accuracy.

(f) Result from highpass f=0.13, lowpass f=0.31 and unfiltered: 91.3% accuracy.

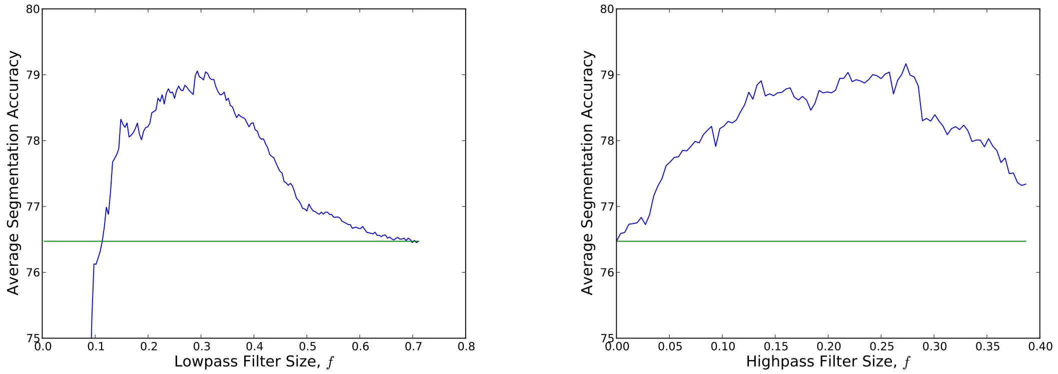
Figure 1. Segmenting a texture mosaic with 0, 1 and 2 filtered images added.



(a) Lowpass filtering.

(b) Highpass filtering.

Figure 2. Effects of low- and high-pass filtering for 50 texture mosaics.



(a) Lowpass Accumulative Filtering with one added filter.

(b) Highpass Accumulative Filtering with one added filter.

Figure 3. Low- and high-pass Accumulative Filtering with a single added filter with a varying filter size, f. The green line represents the segmentation result for the original unfiltered images.

chosen for the segmentation. All segmentation for AF is done using the uniform LBP with P=8 and R=1 where the points are the values of the eight boundary pixels in a 3x3 grid. This is because we found this arrangement gave the best results.

IV. Results

Local Binary Patterns (LBP) have been selected as a standard state-of-the-art operator and there is rich literature on the LBP and its developments along with comparisons to other approaches [7]. We have compared our Accumulative Filtering approach against the standard LBP and best multi-resolution LBP configuration from [5]. Two sets of 50 mosaics each of size 512x512 pixels were generated by random selection of four textures from a subset of 30 textures from the VisTex database [6], one of which is shown in Figure 1(a). This database was chosen because it provides texture images representative of real world conditions. From each texture sample used, one quarter was included in the first mosaic set, a different quarter was used for the second mosaic set and a third quarter was used for the training data for supervised segmentation. The optimum filter sizes for Accumulative Filtering (AF) to use for the Vistex database were learned using the first set of mosaics. The second set of mosaics was then segmented using these filters to demonstrate the effects of the AF process. The selected filter sizes reflect integer values for the expression $f \cdot w$ in Equation 3 for the images used.

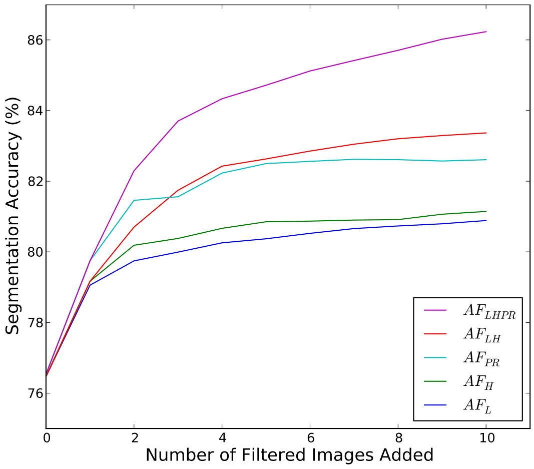


Figure 4. Average mosaic segmentation accuracy during the AF training process.

For the lowpass filters, we used 182 filter sizes between 3.9×10^{-3} and 7.1×10^{-1}. Larger lowpass filter sizes removed no further information from the image and had

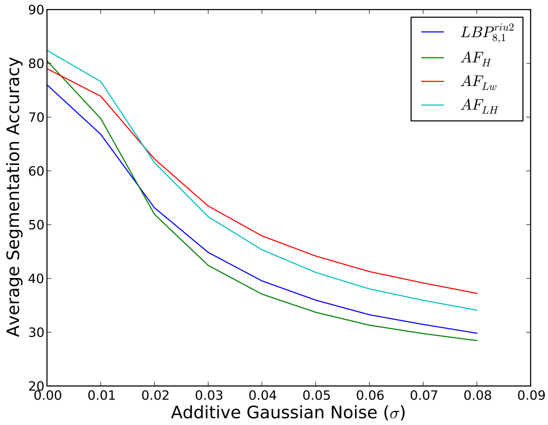


Figure 5. Average mosaic segmentation with increasing levels of additive Gaussian noise.

Table I
AVERAGE MOSAIC SEGMENTATION ACCURACY USING AF AND THE BEST MULTI-RESOLUTION LBP CONFIGURATIONS FROM [5].

Algorithm	Training (%)	Test (%)
$LBP_{8,1}^{riu2}$	76.5	76.0
$LBP_{8,1+24,3}^{riu2}$	75.1	74.8
$LBP_{16,2+24,3}^{riu2}$	70.0	70.2
$LBP_{8,1+16,2+24,3}^{riu2}$	75.9	76.0
AF_L	80.9	79.0
AF_H	81.1	80.5
AF_{LH}	84.0	82.8
AF_{PR}	82.6	81.9
AF_{LHPR}	86.2	85.2

the same effect as no filtering. For highpass, 100 filters sized between 3.9×10^{-3} and 3.9×10^{-1} were used. Further increases to the highpass filter size removed so much information from the image that their inclusion was always detrimental to the process. Initially we performed three experiments on this data to determine the best combination of filters to include; a lowpass test, AF_L, where the filtered images for AF were selected from the 182 lowpass filters, a highpass test, AF_H, where the training process selected from the 100 highpass filters and a combined low and highpass test, AF_{LH}, where the training process was free to choose from any of the lowpass or highpass filters. For each experiment, we determined the optimum filter sizes to use by adding them in stages, singly. The set of filters used at stage i is calculated by:

$$A_i = \{A_{i-1} + f_{max}\} \tag{6}$$

where

$$f_{max} = \operatorname{argmax}_f \left(AF(\{A_{i-1} + f \mid f \in Z\}) \right) \tag{7}$$

and Z is the set of all available filter sizes and $AF(A)$ is the segmentation accuracy from the Accumulative Filtering process using set of filter sizes A. Set A_0 contains only a highpass filter of size 0, which is equivalent to no filtering and segmentation using this is the same as using the standard LBP method, $LBP_{8,1}^{riu2}$. In total, 10 filters were added in each experiment. Statistical analysis using a paired T-test has demonstrated that the accuracy increase with each additional filtered image result added is statistically significant up to and including the tenth additional filter. There were small differences between the results, shown in Figure 4, between AF_L and AF_H at all stages, with AF_H performing slightly better at 81.1% compared to AF_L's 80.9% at 10 added filters. The combined AF_{LH} exceeded the results of both other tests at all stages, with a segmentation accuracy of 83.4% achieved at 10 added filters. The combined experiment showed that the optimum configuration was an equal amount of low- and high-pass filters, and the filter sizes automatically selected by the process were the same or similar to those chosen in the separate experiments. The filter sizes selected during this training process were used in the segmentation of the second set of mosaics; the results, shown in Table I, show that significant improvements can be made in the segmentation results by using AF compared with the standard LBP and multi-resolution LBP configurations.

In [5], Ojala *et al.* found that using multiple LBP operators with different *P* and *R* values gave better results than using a single operator. However, the results do not go beyond combining more than three operators together and only a very limited selection of P and R combinations are used. To better compare this approach against using AF_{LH}, we have used the AF process with varying values of *P* and *R*, AF_{PR}. We used all combinations of *P* values between 8 and 40 (increments of 4) and *R* values between 1 and 5 (increments of 0.5). AF_{PR} achieved an accuracy of 82.6% at when 10 operators were added; better than either AF_L or AF_H. AF_{PR} performed slightly better than AF_{LH} up to 3 additional filters, but AF_{LH} is much better for all subsequent stages tested. Finally, we tried AF selecting from lowpass, highpass and P and R values, AF_{LHPR}. This vastly outperformed all other tests, achieving 86.2% with ten additions. These results demonstrate that low and highpass filtering are best used in conjunction with varying P and R. We segmented an image of a pyramid from the Berkeley Segmentation Dataset (BSDS) [3] using $LBP_{8,1}^{riu2}$, AF_{LHPR} and the two best multi-resolution LBP configurations. The results, shown in Figure 6 show a marked improvement, particularly around the texture boundaries and in homogeneity within the texture regions using our new method . There was also an increase in performance over the multi-resolution LBP segmentations. We also segmented a second BSDS image, shown in Figure 7. The advantages for Accumulative Filtering are clear for this image, with non-trivial improvements over the original and multi-resolution LBP. This shows that while the configurations for the multi-resolution LBP worked well for the pyramid, they only provide a marginal improvement for the chick image. The advantage of Accumulative Filtering is that filter sizes can be learnt for different types of images and so an optimal solution can always be found.

A. Additive Noise

Susceptibility to noise is often considered to be a problem with highpass filtering, however we are pleased to note that this is not the case here. For this analysis, we have introduced additive Gaussian noise to the test mosaics before filtering and segmentation. The filter sizes used for the Accumulative Filtering process are the same as those used in previous tests; learnt from the training mosaics. The texture samples used in segmentation are the original ones and do not include the additive noise. The first test is performing standard LBP histogram segmentation with varying levels of noise added to the mosaics; this is to ascertain the effects of noise on the LBP process absent the effects of filtering. As shown in the graph in Figure 5, noise has a large effect on the standard LBP. Because the texture samples the mosaics are compared to do not include additive noise, the mosaics will have a greater similarity to the texture class with the greatest proportion of the high frequency components that closely resemble Gaussian noise. This will result in a reduced likelihood of the correct texture class being chosen for each pixel and ultimately a lower segmentation accuracy. The segmentation accuracies for highpass Accumulative Filtering are initially higher than those without filtering, but after the noise level has reached $\sigma = 0.02$ it performs slightly worse. This reduction in performance is not as large as could be expected. Lowpass Accumulative Filtering performs better under noise than LBP with no filtering. This is to be expected as the filters will remove much of the noise from the image prior to segmentation. Accumulative Filtering using both low- and high-pass filtering performs significantly better that LBP with no filtering, regardless of the level of additive noise. As such, it appears that the combination of highpass and lowpass Accumulative Filtering is an optimal choice for image segmentation and noise does not markedly affect either type of filter.

V. Conclusions

In this paper a new scale based technique has been presented which increases the segmentation accuracy for textured images using any texture operator which provides a feature vector. Local Binary Pattern histograms have been used as an example to demonstrate the effects of our new process, which we call Accumulative Filtering. During the segmentation process we extract LBP histograms focussing on micro- and macro-structures within the image by using a novel combination of lowpass filters, highpass filters and varying sizes of the LBP operator. Combinations of these histograms give vastly improved results over segmentation using the standard LBP operator alone. We employed a learning algorithm to select the optimum filter sizes to use on a set of mosaics and tested these using a different set. The results show that application of the new process using the learnt filter sizes gives excellent results exceeding those obtained using the best multi-resolution LBP configuration and the standard LBP operator by around 10 percentage points. In addition, segmentation along texture boundaries is improved with the new method and the susceptibility to noise is no different to that of the standard LBP without filtering.

References

[1] Y. He, N. Sang, and C. Gao. Pyramid-based multi-structure local binary pattern for texture classification. In *Proc. ACCV*, pages 1435–1446, 2010.

[2] T. Mäenpää, M. Pietikäinen, and T. Ojala. Texture classification by multi-predicate local binary pattern operators. In *Proc. ICPR*, volume 15, pages 939–942, 2000.

[3] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. ICCV*, volume 2, pages 416–423, 2001.

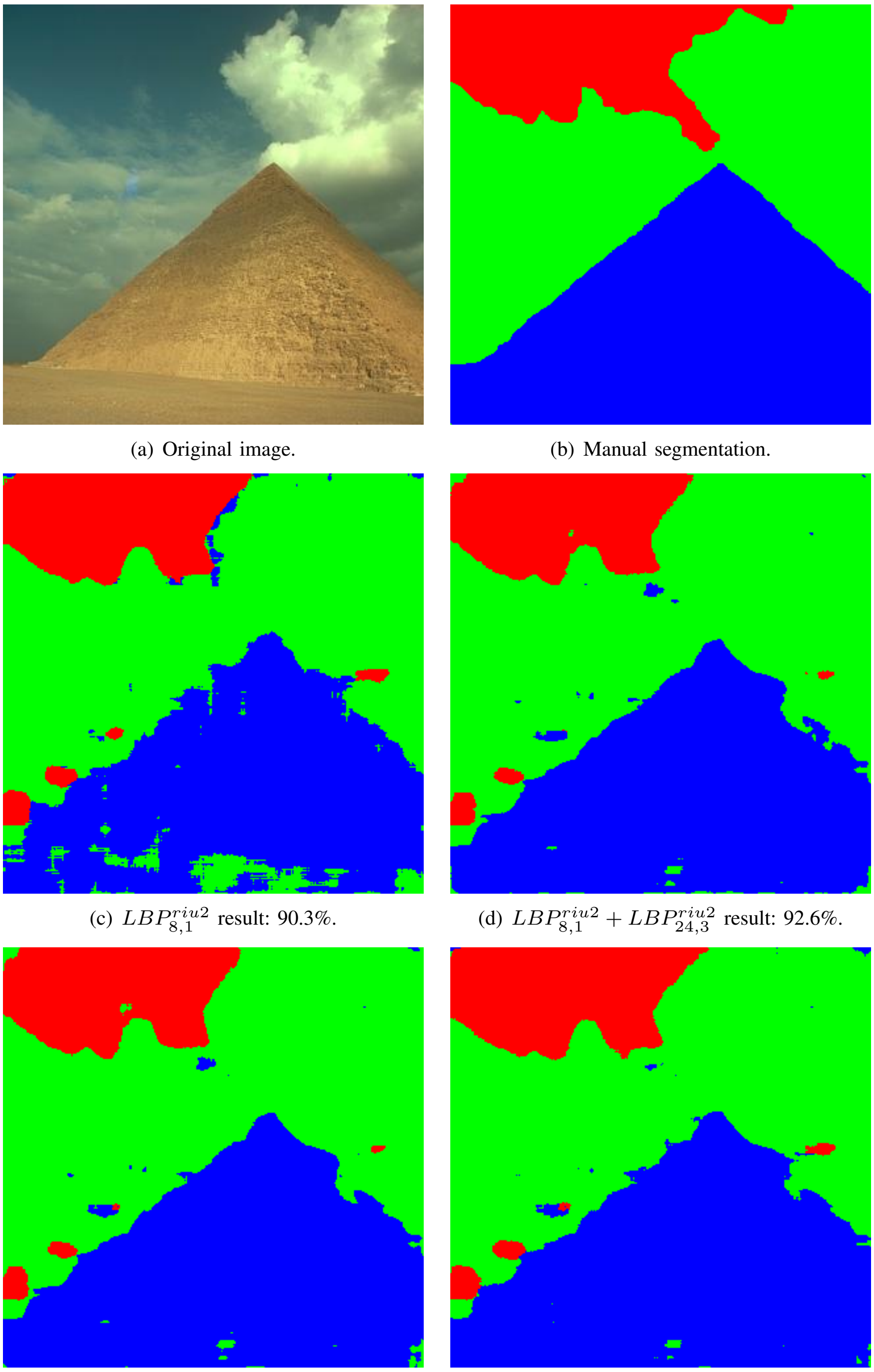
[4] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Patt. Recog.*, 29(1):51–59, 1996.

[5] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI*, 24(7):971–987, 2002.

[6] R. Pickard, C. Graszyk, S. Mann, J. Wachman, L. Pickard, and L. Campbell. Vistex database, 1995.

[7] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen. *Computer Vision Using Local Binary Patterns*, volume 40. Springer Verlag, 2011.

[8] M. Turtinen and M. Pietikäinen. Contextual analysis of textured scene images. In *Proc. BMVC*, pages 849–858, 2006.



(a) Original image.

(b) Manual segmentation.

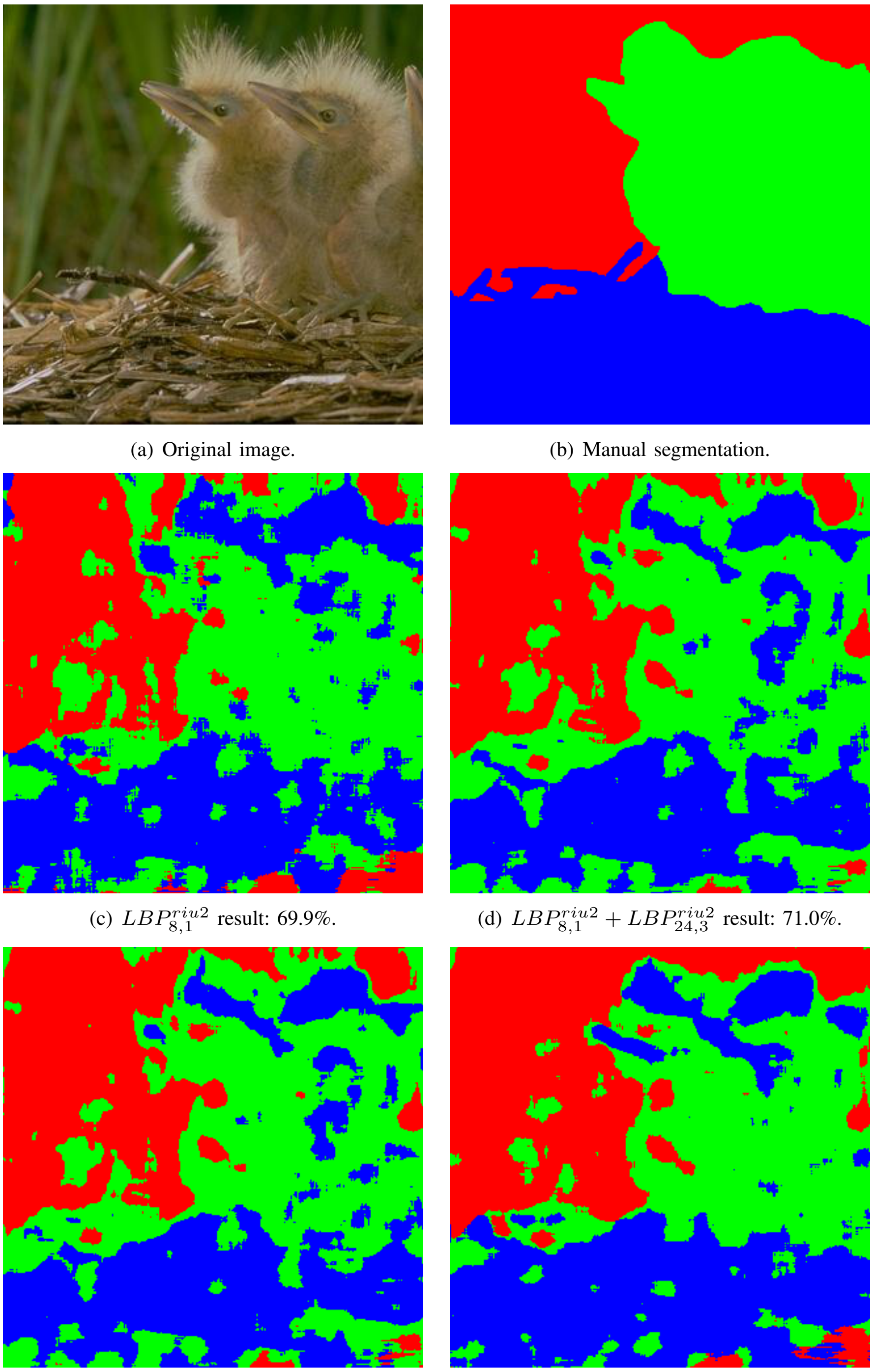
(c) $LBP_{8,1}^{riu2}$ result: 90.3%.

(d) $LBP_{8,1}^{riu2} + LBP_{24,3}^{riu2}$ result: 92.6%.

(e) $LBP_{8,1}^{riu2} + LBP_{16,2}^{riu2} + LBP_{24,3}^{riu2}$ result: 92.7%.

(f) AF_{LHPR} result: 93.4%.

Figure 6. Segmenting an image of a pyramid using Accumulative Filtering.



(a) Original image.

(b) Manual segmentation.

(c) $LBP_{8,1}^{riu2}$ result: 69.9%.

(d) $LBP_{8,1}^{riu2} + LBP_{24,3}^{riu2}$ result: 71.0%.

(e) $LBP_{8,1}^{riu2} + LBP_{16,2}^{riu2} + LBP_{24,3}^{riu2}$ result: 72.4%.

(f) AF_{LHPR} result: 80.4%.

Figure 7. Segmenting an image of chicks in a nest using Accumulative Filtering.