COLOUR TEXTURE SEGMENTATION USING EVIDENCE GATHERING

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

Keywords: Colour, Texture, Segmentation, Evidence Gathering, LBP

Abstract

A new approach to colour-texture segmentation is presented which uses Local Binary Pattern data and a new colour quantisation scheme based on hue and saturation to provide evidence from which pixels can be classified into texture classes. The proposed algorithm, which we contend to be the first use of evidence gathering in the field of texture classification, uses Generalised Hough Transform style R-tables as unique descriptors for each texture class. Tests on remotely sensed images demonstrate the superiority of the colour-texture algorithm compared to the established JSEG algorithm; a notable advantage of the new approach is the absence of over-segmentation. The VisTex database is used to compare the colour-texture algorithm with alternative methods, including its grey-scale equivalent, for the segmentation of colour texture images; providing good results with smooth texture boundaries and low noise within texture segments.

1 Introduction

Remotely sensed images contain a variety of colour and texture information representing many different features, each formed of its own unique blend of patterns. The images of the Earth's surface captured by satellites are used for many applications; leading to conclusions about the rate of coastal erosion, deforestation and urban development within a region. As it is an appropriate focus for the application of texture segmentation, we require a technique to divide the image into regions based on colour-texture features.

There are three main ways of combining colour and texture information into a single operator. The first is a parallel combination, whereby colour and texture operators are applied separately to an image and the results are concatenated into a single feature vector. The advantage of this approach is that colour information can easily be added to an existing texture algorithm by applying a colour operator in parallel. The second approach is sequential, wherein the colour operator is applied first and the texture algorithm operates on this "colour-space"; finding texture within the colour. The advantage of this method is that the feature vector provided by the algorithm can still be processed in the same way as that obtained from a pure texture version. An example of a sequential colour-texture operator is the JSEG algorithm developed by Deng and Manjunath [2]. The final way of combining colour and texture information is the integrated approach. This involves fusing colour and texture to form a single feature vector.

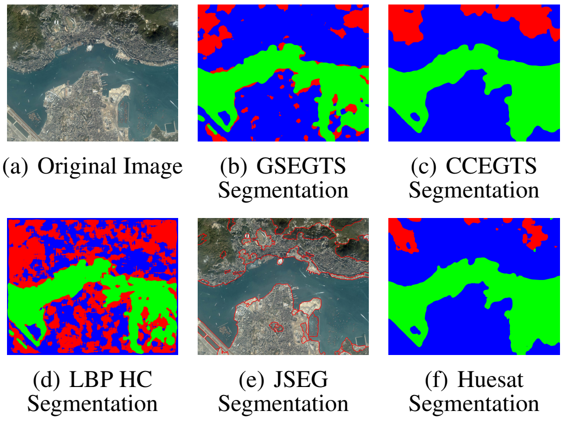

(a) Original Image (b) GSEGTS Segmentation (c) CCEGTS Segmentation (d) LBP HC Segmentation (e) JSEG Segmentation (f) Huesat Segmentation

Figure 1: Segmenting an image of Hong Kong with new and comparison algorithms.

Mäenpää and Pietikäinen [3] claimed that using colour and texture in parallel is not the most effective way of utilising the information and suggested that under static illumination conditions colour alone works best, while grey scale alone works best under varying illumination. However, Palm [8] showed that adding colour histogram information to grey scale features in a parallel manner gave better results for texture classification than the grey scale operator alone. He further claimed that using an integrated colour texture feature can yield an even better result.

We describe a new integrated approach to colour-texture segmentation which determines texture class through evidence gathering, a process well known for its noise reduction properties. The evidence is collected in the form of a Local Binary Pattern (LBP) code [7] and colour class for each pixel, which are stored in Generalised Hough Transform (GHT) [1] style R-tables. Accumulators are used to store votes for each texture class and determine the final classification for each pixel. This new approach is the first use of evidence gathering to determine texture and has been demonstrated to give appealing results for texture segmentation, as illustrated by Fig. 1, while maintaining smooth texture boundaries and minimising noise. This algorithm is referred to henceforth as the Colour Class Evidence Gathering Texture Segmentation (CCEGTS) algorithm. We also introduce a new colour quantisation scheme called Huesat which is based on hue and saturation and provides the colour classes for our CCEGTS algorithm. Our evidence gathering technique can be used in the absence of colour information, referred to as the Grey Scale Evidence Gathering Texture Segmentation (GSEGTS) algorithm, and this is used to demonstrate the additional benefits gained by including colour in the segmentation. We demonstrate that CCEGTS provides consistently better segmentation results than GSEGTS on a texture database and on satellite images, as shown in Fig. 1. Since CCEGTS and GSEGTS both use uniform LBP to provide texture information we compare our algorithm against LBP histogram comparison, which performs segmentation based on the statistical distribution of LBP codes within the image. This comparison demonstrates that the inclusion of structural information in the evidence gathering approach benefits texture segmentation. Additionally, we show that our approach gives better results than RGB histogram comparison, which is a pure colour operator, to demonstrate that combining colour and texture information into a single operator performs better than either feature on their own. We apply these algorithms to remotely sensed images to demonstrate the advantages in segmentation of our new approach.

2 Evidence Gathering Texture Segmentation

A new evidence gathering approach to texture segmentation is proposed which uses the principles of template matching present in the Generalised Hough Transform (GHT) [1] and modifies it to match texture instead of shape. The technique exploits a property of the Local Binary Pattern (LBP) [7] texture descriptor which is that if there is structure in the image space, there must be structure in the LBP space. Mäenpää and Pietikäinen observed in [4] that each LBP code limits the set of possible codes adjacent to it. This implies that the arrangement of LBP codes within a texture is not random and that taking a histogram of the codes reduces the available information further to that originally lost in the LBP process. It is possible for several textures to have the same histogram, rendering such methods incapable of distinguishing between them. By storing the LBP code along with its offset to the centre of the texture region for each pixel as an entry into an R-table, this structural information is not lost and a unique descriptor is produced which can be used in the classification and segmentation of images. The descriptor is unique because it can be used to regenerate the array of LBP codes that represent the texture sample, unlike a histogram of LBP codes. The new algorithm is the first use of evidence gathering in texture segmentation and achieves high efficiency by transferring the principles of low computational complexity present in the GHT method to texture analysis.

The idea of segmenting an image by combining colour and texture information in an integrated manner can be applied to the evidence gathering approach by indexing each R-table entry by colour class as well as the LBP code. This combines colour information with texture and helps to reduce votes for incorrect textures by limiting voting to within colour classes. Our new Huesat colour quantisation algorithm is applied to the image to assign each pixel into a colour class. The colour classes are determined from the hue and saturation calculated from the RGB values of each pixel. Other colour quantisation approaches could also be used.

2.1 Local Binary Patterns

Local Binary Patterns are texture descriptors which label individual pixels in an image with a code corresponding to the local texture pattern surrounding the pixel. First introduced by Ojala *et al.* [6], the earliest form of LBP used the centre pixel of a 3 by 3 grid, g_c, to threshold each of the eight neighbouring pixels g_0 to g_7. This produced an eight bit binary code which represents the texture element present at that point. The LBP was later extended to give the texture pattern for *P* points on a circle of radius *R*. It was observed that certain fundamental patterns make up the majority of all LBP patterns observed [7]. These were found to be the patterns which had at most two zero to one transitions ($U(LBP_{P,R}) \leq 2$) and are called *uniform* LBP patterns. All of the uniform patterns are labelled according to the number of '1' bits in the code. When *P* is equal to eight, there will be ten different patterns: the uniform patterns from '0' to '8' and the pattern '9' which is the agglomeration of all other patterns (for which $U(LBP_{P,R})$ > 2). Since the sampling positions for the neighbouring points are arranged in a circle, the uniform LBP is, by nature, rotation invariant. This is because if the sample image texture is rotated, the LBP code produced will still have the same number of zero to one transitions and the same number of '1' bits, resulting in an identical uniform LBP code regardless of the order of the bits. The rotation invariant uniform LBP code for a point, $LBP_{P,R}^{riu2}$, is calculated by:

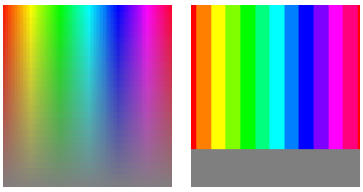
$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{if} \quad U(LBP_{P,R}) \leq 2 \\ P+1 & \text{otherwise} \end{cases} \tag{1}$$

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.

The values of *P* and *R* can be adjusted to fit the requirements of the application. A greater number of points increases the resolution of the texture descriptor, allowing a higher level of differentiation between different texture patterns. Accordingly, increasing *P* also increases the number of different LBP codes. The value of the radius determines



(a) Palette (b) Quantised Palette

Figure 2: Colour palette before and after quantisation.

the scale of the LBP. If R is high, the LBP code will be representative of the texture pattern over a large area. Similarly, if R is low, the LBP algorithm will use a smaller area of the image to determine the texture element.

2.2 Colour Quantisation

We have developed a new colour quantisation scheme based on hue and saturation to assign each pixel into a colour class. The hue spectrum is quantised into twelve 30 degree intervals, each of which is assigned a colour class number. In addition, a thirteenth class is created for colours with a saturation of less than 25%. Each colour class is intended to represent a group of colours recognisable under a single label such as "red", "pink" or "purple". The low saturation class is intended to capture grey pixels. Colours under this condition can have small visual differences but large differences in hue, so it is important to assign them their own class. Eqn. 3 shows the calculation of colour class from hue and saturation. The hue values are offset by 15 degrees to ensure that the primary colours fall in the centres of their respective colour classes. The effects of this colour quantisation scheme are illustrated in Fig. 2, showing a smoothly varying palette categorised into regions of colour.

$$C = \begin{cases} \text{trunc}\left(\frac{hue+15}{30}\right) & \text{if} \quad sat \geq 0.25 \\ 12 & \text{otherwise} \end{cases} \tag{3}$$

Twelve colour classes are chosen because this number means each class is small enough to exploit class separation, but large enough to ameliorate noise. The new colour quantisation scheme, referred to as Huesat, can be used on its own as a colour segmentation algorithm by applying the same principles of histogram comparison as used by the RGB histogram comparison algorithm in [10]. This is shown in Eqn. 4 where n is the number of colour classes, I is the image histogram and M the model histogram.

$$H(I, M) = \frac{\sum_{j=1}^{n} \min(I_j, M_j)}{\sum_{j=1}^{n} M_j} \tag{4}$$

2.3 Evidence Gathering

As with the GHT, before sample images can be analysed an R-table must be generated for each known texture class. This describes the structure and composition of a section of the texture and is used to classify the texture class of the sample images. Sub-images, or cells, are taken from the training images and the LBP code and colour class are calculated for each pixel within the cell. The R-table contains a number of bins equal to the number of different LBP codes that exist for the version of the LBP that is being used multiplied by the number of colour classes. For LBP P values of eight and standard colour quantisation giving thirteen colour classes, the number of bins will be 130; thirteen bins for each of the nine uniform LBP codes and thirteen bins for all other LBP codes which are not classified as one of the uniform patterns. For each pixel in the cell an entry is submitted to the bin corresponding to the LBP code and colour class for that pixel. The entry is a two dimensional vector **r**=(x_r,y_r) representing the translation from the pixel to the reference point of the cell. This reference point is usually chosen to be the centre. In Fig. 3, the top left pixel in the cell has an LBP code of '1' and the colour class is orange, so an entry is made in the '1,0' bin with the vector (2,2) which maps the top left pixel to the centre. The size and number of cells taken from the training images are not fixed and these parameters can be tailored for different applications. The size of the cell should be sufficiently large to contain at least one full example of the repeating pattern in the texture. Having multiple cells for each texture class will provide more evidence for classification during the segmentation process.

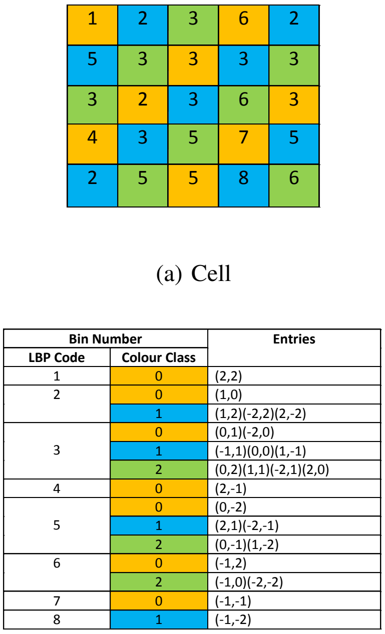
The following equation is used to calculate the R-table entry for each pixel **x** = *(x,y)* in a cell of centre **c** = (x_c, y_c):

$$\mathbf{r} = \mathbf{c} - \mathbf{x} \tag{5}$$

where the R-table index is the LBP code calculated by Eqn. 1 and the colour class calculated by Eqn. 3 at point **x** = *(x,y)*. The grey scale version of the algorithm can be easily achieved by setting the colour class of each pixel to 12 (grey) regardless of the actual colour. This effectively means that each entry is indexed only by the LBP code.

2.4 Voting

Evidence is stored in an accumulator array and a separate accumulator is maintained for each of the texture classes that are being searched for. In the segmentation of sample images, the LBP code and colour class for each pixel in the entire image is calculated. The entries in the R-tables represent the possible locations of the current pixel relative to the reference point of the cell. For the example in Fig. 3, if a pixel in the sample image had an LBP code of '3' and colour class '1' (blue), it could correspond equally to any of the three positions within the cell also with that combination of LBP code and colour class. For each in turn, votes are made for the area that would cover the entire cell positioned on that pixel. Rephrasing the GHT, the algorithm becomes: for each pixel **x** in the image, increment all the corresponding points in a cell centred on the point **x** + **r** in the accumulator array A where **r** is a table entry indexed by the LBP code and colour class at point **x**. Maxima in A correspond to possible instances of the texture T.



1	2	3	6	2
5	3	3	3	3
3	2	3	6	3
4	3	5	7	5
2	5	5	8	6

(a) Cell

Bin Number		Entries
LBP Code	Colour Class	
1	0	(2,2)
2	0	(1,0)
	1	(1,2)(-2,2)(2,-2)
3	0	(0,1)(-2,0)
	1	(-1,1)(0,0)(1,-1)
	2	(0,2)(1,1)(-2,1)(2,0)
4	0	(2,-1)
5	0	(0,-2)
	1	(2,1)(-2,-1)
	2	(0,-1)(1,-2)
6	0	(-1,2)
	2	(-1,0)(-2,-2)
7	0	(-1,-1)
8	1	(-1,-2)

(b) R-table

Figure 3: Example LBP and colour class values for a 5x5 pixel cell and corresponding R-table. The reference point of the cell is the centre pixel with LBP code '3' and colour class blue. Empty R-table bins are not shown.

Voting is done in blocks rather than for individual pixels because texture covers an area and a single pixel on its own does not contain texture. The area covered by each block vote is equal to the area of the cell from which the evidence was gathered. The three block votes for an LBP code of '3' and colour class '1' using the R-table in Fig. 3(b) are shown in Fig. 4. The algorithm is effectively searching the sample image for the texture structure observed in the training cell. In Fig. 4, it can be seen that nine of the pixels in the image were within all three possible cells for that R-table and hence these pixels have a higher probability of belonging to that texture class. Compared to the GSEGTS algorithm, the computational cost for CCEGTS is reduced since there will be fewer block votes made for each pixel in the image since the entries in each grey-scale R-table bin are spread over a number of bins in the new colour version of the algorithm.

The R-table can also be extended to include the LBP values of the eight surrounding pixels as well as the vector as the R-table entry. When votes are calculated for the sample image, an extra vote is awarded for each neighbouring pixel in the sample image whose LBP code matches its corresponding pixel included in the R-table entry. This gives a maximum of nine block votes that can be awarded for each R-table entry and works on the principle that the higher the number of neighbouring pixels that match a texture class, the higher the chance of the centre pixel belonging to that texture class.

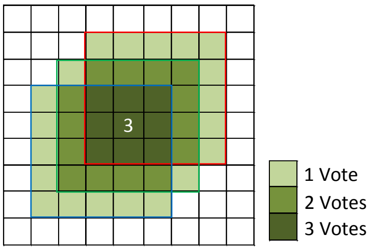


Figure 4: Accumulator showing block votes for three R-table entries, bordered by red, green and blue rectangles.

An accumulator for each texture class maintains the number of votes for each pixel for that texture. If there is more than one cell for a texture class, the votes of the subsequent cells are added to the accumulator for the first cell. When the voting process is finished, the higher the number of votes for each pixel, the higher the probability of the pixel belonging to that texture class. It is important to note that analysis of a single pixel yields evidence for many other pixels. This works because if there is structure in the texture, the LBP code at a point is related to those around it. Using a higher number of cells per texture class increases the amount of evidence used to classify pixels and leads to a higher accuracy. Segmentation is performed by filling an accumulator for each texture class and assigning each pixel to the texture class with the highest number of votes at that point.

2.5 Normalisation

It can be observed that different textures have different voting strengths. This means that some textures could give a larger number of votes for an incorrect texture than another texture could give for a correct match. This leads to cases where votes from one texture overpower those from another, distorting the segmentation results. A solution is to normalise the voting, whereby the votes from each texture are weighted according to their strength. One way of calculating the strength is to add up the total number of votes for the texture over the entire image and divide by the number of pixels. When all votes for a texture are divided by its strength factor the 'stronger' textures will have their influence over the regions of other textures weakened, reducing the 'overspill' effect. The equation for performing normalisation on an accumulator A of size w by h is:

$$A_{norm}\left(x,y\right) = \frac{A\left(x,y\right)\cdot w\cdot h}{\sum_{a=0}^{w}\sum_{b=0}^{h}A(a,b)} \tag{6}$$

If normalisation is required where one texture is 'weaker' than the others, its use can restore the texture boundaries to their correct locations.

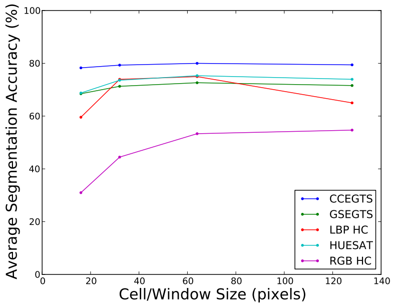


Figure 5: Comparison of performance between new and existing algorithms at various cell/window sizes. Each point is the average result from 50 mosaic segmentations.

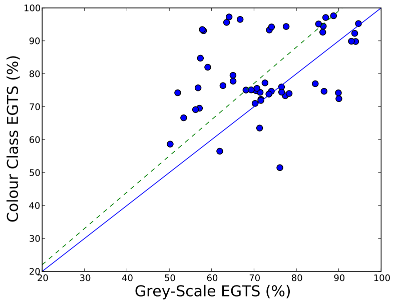


Figure 6: Comparison between CCEGTS and GSEGTS with three cells of size 64x64.

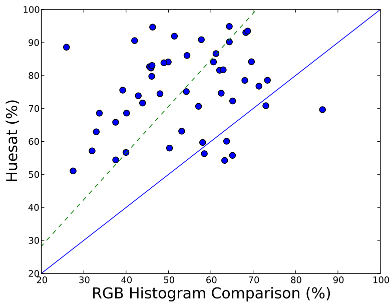


Figure 7: Comparison between Huesat and RGB Histogram Comparison using windows of size 64x64.

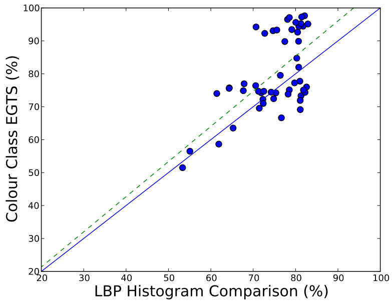


Figure 8: Comparison between CCEGTS with three cells of size 64x64 and LBP Histogram Comparison using windows of size 64x64.

3 Results

3.1 Texture Mosaics

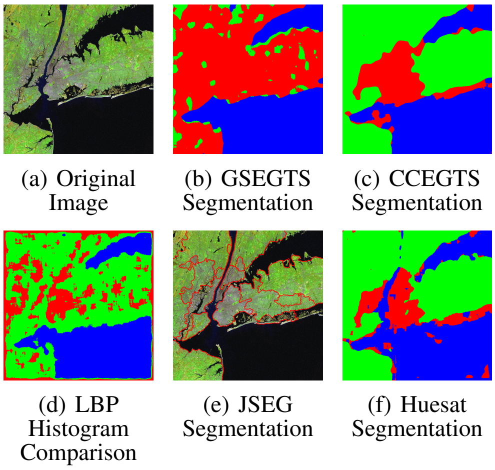
To test our CCEGTS algorithm, a set of fifty mosaics was generated by random selection from a subset of 30 textures from the VisTex database [9], which consists of real world colour texture images. For comparison, segmentations were also run using GSEGTS, LBP histogram comparison [5], RGB histogram comparison [10] and our Huesat colour quantisation. Results from the tests conducted at four window sizes are shown in Fig. 5 and comparisons between individual algorithms are shown in Figs. 6, 7 and 8 where the solid blue line represents the line of equality and the dashed green line represents the trend. Each dot in the scatterplot represents the segmentation of a single mosaic under both algorithms being compared.

Tests against the grey-scale EGTS algorithm are important to show that inclusion of colour information has a positive effect on the performance of the algorithm. We have found that the GSEGTS performs significantly better than LBP histogram comparison on a subset of the Brodatz database, however the two algorithms gave similar results for the VisTex subset used in this evaluation. A significant advantage was noted for GSEGTS for cell sizes of 16 and 128, however histogram comparison slightly outperformed it for cell sizes of 32 and 64. This can be attributed to the greater proportion of non-uniformly structured textures present in the VisTex subset than in the Brodatz subset. As described in Section 2, the evidence gathering method works by recognising the structure of texture elements within an image. For textures that do not adhere to a rigid structure, such as the bark images included within the VisTex subset, a performance decrease can be expected.

RGB colour histograms provide a pure colour algorithm to demonstrate the advantages of our new Huesat quantisation scheme. The result for Huesat is very good and Fig. 7 shows the huge improvement it offers over the RGB quantisation. Additionally, the integrated combination of colour and texture provided by our colour class EGTS algorithm provides better results than either of the pure texture or colour operators used. This demonstrates that while colour or textural information on its own can provide a satisfactory solution to image segmentation, the integration of colour information with textural features in our CCEGTS algorithm gives notably improved results, while simultaneously reducing execution time.

3.2 Remote Sensing

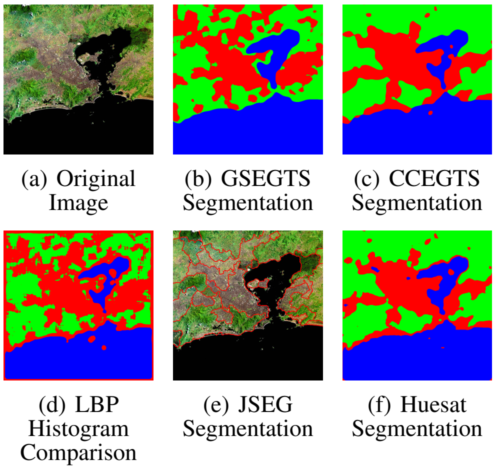
We also applied our new approach to the segmentation of a number of remotely sensed images. Segmentation using CCEGTS and GSEGTS requires the provision of samples of the texture classes to be found in the image prior to segmentation. These are supplied to the algorithm by entering the coordinates of a location within the image containing that texture class. The example shown in Fig. 1 shows an aerial view of Hong Kong, containing urban,



(a) Original Image (b) GSEGTS Segmentation (c) CCEGTS Segmentation

(d) LBP Histogram Comparison (e) JSEG Segmentation (f) Huesat Segmentation

Figure 9: Segmenting an image of New York with new and comparison algorithms.



(a) Original Image (b) GSEGTS Segmentation (c) CCEGTS Segmentation

(d) LBP Histogram Comparison (e) JSEG Segmentation (f) Huesat Segmentation

Figure 10: Segmenting an image of Rio de Janeiro with new and comparison algorithms.

non-urban and water sections. An improved segmentation result was achieved using our new CCEGTS algorithm when compared with the GSEGTS algorithm. In particular it can be seen that the urban area in the lower half of the image contains some noise in the GSEGTS segmentation which is completely eliminated using the colour version. Comparisons against RGB histogram comparison, LBP histogram comparison and JSEG segmentation also indicate the superiority of our new approach. In an image such as this which contains strong colour differentiation between semantic sections of the image it can be noted that our CCEGTS segmentation is a less noisy alternative to pure colour RGB histogram comparison. The unsupervised segmentation result obtained using the JSEG algorithm performs less favourably due to the high rate of oversegmentation. The inclusion of colour information is especially important in the image of New York in Fig. 9 as both GSEGTS and LBP Histogram Comparison give poor results. CCEGTS and Huesat give more visually appealing results with CCEGTS again providing a smoother, less noisy segmentation than Huesat alone. Fig. 10 highlights again the problems of oversegmentation with the JSEG algorithm, as the city is divided into many regions. CCEGTS assigns most of the city region into a single contiguous block; a useful property for later analysis of the results.

4 Conclusions

We have presented a new method for image texture segmentation which we contend to be the first use of an evidence gathering approach in the field of texture analysis. In contrast to conventional methods which compare measurements from a sample of an image to training data to classify a single pixel, our approach compiles information gathered from each pixel into evidence to support the classification of nearby pixels into each known texture class. We have also presented a new colour quantisation scheme based on hue and saturation which is integrated into the texture operator. Our colour class EGTS algorithm uses the structure of texture and colour within an image for segmentation. We have tested the theory of using evidence gathering for texture by conducting segmentations on a subset of the VisTex database; demonstrating superiority of our CCEGTS algorithm when compared to the basic operators from which the evidence gathering method builds upon. When applied to satellite imagery CCEGTS provides appealing segmentations which are a substantial improvement on the GSEGTS algorithm. This, alongside the results obtained from VisTex, shows that the inclusion of colour information provides a better result at a decreased computational cost than texture information alone.

References

[1] D.H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Patt. Recog.*, 13(2):111–122, 1981.

[2] Y. Deng and BS Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE TPAMI*, 23(8):800–810, 2001.

[3] T. Mäenpää and M. Pietikäinen. Classification with color and texture: jointly or separately? *Patt. Recog.*, 37(8):1629–1640, 2004.

[4] T. Mäenpää and M. Pietikäinen. Texture analysis with local binary patterns. *HPRCV*, pages 197–216, 2005.

[5] 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.

[6] 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.

[7] 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.

[8] C. Palm. Color texture classification by integrative co-occurrence matrices. *Patt. Recog.*, 37(5):965–976, 2004.

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

[10] M.J. Swain and D.H. Ballard. Indexing via color histograms. In *Proc. ICCV*, pages 390–393, 1991.