https://doi.org/10.2352/ISSN.2169-2629.2018.26.19

Recovering a Color Image from its Texture

Graham Finlayson, Seth Nixon, University of East Anglia; Norwich, United Kingdom

Abstract

Texture features can be considered as methods for encoding an image: taking pixel intensities or filter responses and forming them into a description which can be used to solve problems including recognition and matching. In this paper we are considering the inverse problem: given a textural representation of an image, how well can we recover the original.

We show how the LBP method encodes comparative relationships between pixels and relative to these relations we can recover an image with our new method. We extend the recovery method to work with the Sudoku texture representation (an extension of LBP). We show that this method produces a reconstruction more correlated with the original image than the prior art.

Introduction

Texture has been studied in computer vision for over 50 years. Important methods that have been developed include co-occurrence representations [1, 2, 3], and representations relevant to human vision including Gabor Filters [4, 5, 6]. Mostly, the goal of a texture representation is to enable recognition however the structural aspect of texture is also important. We might ask, for example, how easy it is to synthesize a texture from an exemplar. An important work here is that of Efros et al. where given a seed image or image patch they "grow" a texture which is visually similar [7]. Particularly relevant to this paper we might ask how much of the original image can we recover from the texture representation.

This last question is interesting. Image processing is full of examples of dual representations. For example it is well known that an image and its Fourier transform are bijectively related. And, of course, some image processing tasks are better performed in one representation than the other (e.g. fast convolution should be carried out in Fourier space) [8]. We propose that a measure of texture "as a representation" is the extent to which it "encodes" the original image.

Image reconstruction from texture has many applications. An example is Cryptography: plausibly an image's texture features could be transmitted and then decoded elsewhere off-line to recover the image. Another example would be analysis of a texture representation. The reconstruction would form a visualization of the knowledge encoded therein.

Local Binary Patterns (LBP) are one of the most successful and commonly used texture representations. In their simplest form an LBP feature is a binary string which encodes whether the brightness at an eccentric pixel is more or less than (1 or 0) than the center. These 1s and 0s are read out to form a binary number which can be read as an integer. Typically these numbers are grouped (e.g. histogrammed) to form a feature vector for indexing. The key strengths of LBP are its invariance to monotonic changes in image brightness (by scaling, gamma functions or tone mapping) and its robustness to rotation [9, 10].

There are many extensions to LBP. One such is the Sudoku representation. Rather than encoding relationships to a central pixel pixels are ranked in a 3x3 neighborhood (using numbers 1 through 9). By construction the Sudoku grid has the same monotonic invariance as LBP but is a richer feature set [11].

Both LBP and Sudoku deliberately remove greater or lesser amounts of magnitude information. This can be considered as removing unimportant intensity representation in favor of a more structural description. The question we ask is: given the per-channel textural information for an image; how well can we recover the color image? We present results for two methods: one previous art known as the Minimum Contrast (MC) algorithm [12] and our own proposed method called Quadratic Reconstruction (QR). We show that essentially the LBP or Sudoku texture encoding (at a pixel) specifies the intensity relationship that pixels in a proximal region need to satisfy. Over the whole image the effect of these local relations propagate. We demonstrate how - using an optimization technique called Quadratic Programming - we can recover the minimum norm image that satisfies the constraints. Compared to the prior art MC method, our new QR method provides a much better recovery.

The rest of this paper is organized as follows: First we will detail the canonical LBP and Sudoku representations and their transformation into feature vectors. Secondly we shall detail the MC and QR algorithms. Finally experiments are presented.

Background

Local Binary Patterns

An LBP is a binary string which describes a neighborhood of pixels. It is formed by comparing a central pixel with its neighbors, if the neighbor is greater than the center it is assigned 1, if lesser it is assigned 0.

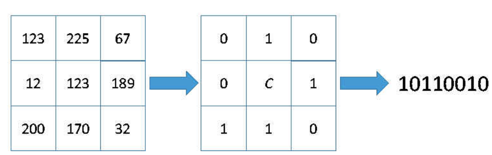

Figure 1. *Transformation of a neighborhood into its LBP representation.*

This number is then read clockwise starting from an arbitrary point. To obtain rotational invariance this number is shifted, with circular wraparound, to its maximum value. For reference the LBP in Figure 1 would be 11001010, or 202 in decimal. This can be expressed as

$$LBP(N_p) = \sum_{i=1}^{P} (p_i > p_c)^{2(i-1)} \quad (1)$$

where N is the neighborhood, P is the number of pixels and p_c is the central pixel. Once this process has been applied to every pixel in an image a histogram of all the resultant integers forms a feature vector. Two textures are deemed to be similar if their histograms match closely to one another. Matching can be performed using a distance metric, e.g. chi-squared [13] or Histogram Intersection [14]. This representation has two main advantages: firstly, it is invariant to any monotonic changes in the gray-scale and secondly it is rotationally invariant. The reader can apply any increasing function to the example given in Figure 1 and observe that the same LBP will result. The LBP feature is also very computationally efficient; the only calculations needed are inequality tests and for an eight pixel neighborhood there are only 36 rotationally invariant 8-bit LBPs.

An important addition to the LBP was the concept of uniformity. Statistically up to 95% of patterns in an image contain two or less 0 to 1 transitions. These LBPs are denoted uniform and are used to form the histogram with the remaining patterns grouped into one bin. This significantly compresses the feature length such that each histogram now has $p+2$ bins, where p is the number of points in the neighborhood.

The Sudoku texture representation

Sudoku patterns are formed in a similar way to LBP however instead of using just the central pixel for comparisons all pixels in a neighborhood are compared simultaneously to form a rank-ordering.

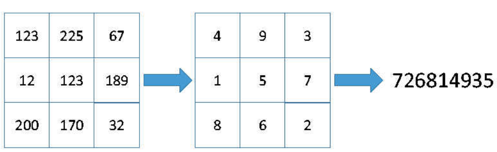

Figure 2. *Transformation of pixel intensities to rank values.*

This is read clockwise as an integer starting at the right-most pixel with the central pixels rank appended to the end.

This can be expressed as

$$S(P) = \sum_{i=1}^{n} RANK(p_i) * 10^i \quad (2)$$

where the *RANK* operator assigns the rank to pixel p in neighborhood P. This number is then shifted with circular wrap around to its maximum value to achieve rotational invariance. The pattern in Figure 2 would be 786139254 and then shifted to 925478613. This process is applied to every pixel in an image to form a "Sudoku" image. Pixel equality is also incorporated into the representation. This allows the description to more effectively account for near uniform neighborhoods. One strategy for implementing equality is to use ranks for the unique values in a neighborhood. So, if 2 pairs of numbers in a 3x3 neighborhood hold the same value then the maximum rank would be 7. And, two of the ranks up to 7 would be repeated (indicating equality).

Empirically in [11] it was found that the probability of different Sudoku numbers was very non-uniform. So, before aggregating the Sudoku numbers over an image each Sudoku number is 'tone mapped' such that the probability is uniform (across the set of all Sudoku numbers). In effect histogram equalization is used as a processing step.

Now, Sudoku numbers are histogrammed as before and the histogram of tone-mapped features is used to drive recognition. In [11] it was shown that the Sudoku representation delivered better texture indexing compared with LBP.

Image recovery

While the methods detailed above go as far as to form feature vectors, we would like to take a step back and consider just the pattern in 3x3 neighborhoods. That is we just use the raw LBP pattern at a pixel. Sometimes these raw patterns are called "Glyphs" and we use this nomenclature here. This *glyph* retains all of the structural information of the neighborhood while still preserving each pixels spatial location. This is important as once the patterns are rotated the positioning of each pixel becomes near-arbitrary making recovery a significantly more complex task. The following two methods, Minimum Contrast and Quadratic Reconstruction, use these glyphs as the basis for their reconstructions. The recovery pipeline is shown in Figure 3.

Minimum Contrast

The minimum contrast algorithm was proposed in [12] as a way of inverting the local binary pattern glyph. The primary goal was two-fold, firstly to provide a recognizable reconstruction of an input image and secondly to show what information was lost in the LBP conversion.

To reconstruct an image using this method first consider the 4x3 neighborhood in Figure 4

If this is deconstructed into its corresponding LBP patterns we end up with 2 glyphs centered around rows, columns 2,2 and 2,3 as in Figure 5.

If we examine first neighborhood a we can see that the top left pixel is less than the center, as defined by the 0. This means that the central pixel of a must be at least 1 pixel intensity greater than the top left. We can also see that the bottom right pixel is greater than the center, as defined by the 1. This means that the bottom right of a is at least 1 pixel intensity greater than the center; and transitively at least 2 pixel intensities greater than the top left. If we then move to examine neighborhood b we can see that the central pixel of b is greater than the bottom central pixel, which corresponds to the bottom right pixel of a, and that the bottom right pixel of b is greater than the center. From this we can deduce that the bottom right of b is at least 5 pixel intensities greater than the top left of a, or in other words the **minimum contrast** between the two pixels is 5. Using this series of greater than relationships we can express this as a path, see Figure 6.

It is worth noting that this is not the longest path available which reaches the bottom right. However this is the longest *explicit* path defined by glyphs a and b. The actual longest path could become explicit if the glyph around row, column (1,1) were

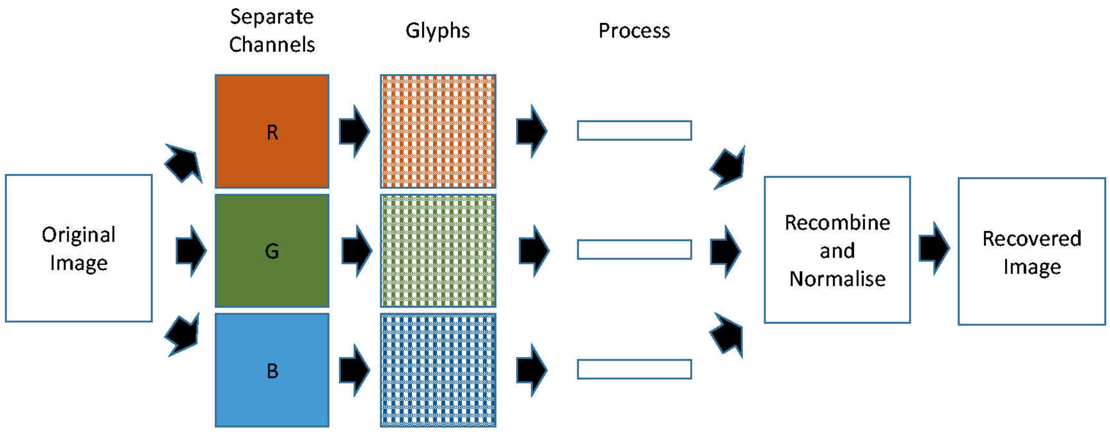

Figure 3. *The image recovery pipeline.*

1	33	85	197
242	69	124	12
145	21	83	211

Figure 4. *A neighborhood of pixels.*

0	0	1
1	$C_{2,2}$	1
1	0	1

0	0	1
0	$C_{2,3}$	0
0	0	1

Figure 5. *LBP Transformations of Figure 4.*

Figure 6. *One greater than path through 4.*

part of this example.

To expand this process we must start at every local minimum in an image and recursively generate every possible greater than path through said image. In each pixel location we store the length of the longest path to that pixel. The path length is defined as the pixel brightness in the reconstructed image. The final result is an image in which the LBPs match exactly to the original.

We extend this process to use the Sudoku feature. In a Sudoku glyph we have the rank ordering of a neighborhood expressed as the numbers 1 to 9. This means that if one pixel is ranked 9 and another pixel is ranked 2 the minimum intensity difference between those two pixels is 7. To incorporate this into the minimum contrast algorithm when we explore a greater than relationship we increment the path length by the absolute rank difference between the two pixels. This produces an image where the Sudoku codes match closely to the original image. For an example of both LBP and Sudoku minimum contrast reconstruction see Figure 7.

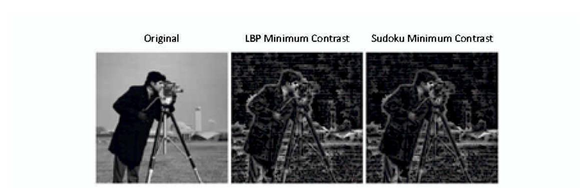

Figure 7. *Examples of LBP and Sudoku minimum contrast reconstruction on MATLABs "Cameraman.tif".*

Quadratic Reconstruction

Our proposed method forms the problem in terms of quadratic programming. If we consider an image to be a vector x with the pixels as variables $x_1 \ldots x_n$. For LBP we can formulate a set of linear constraints $c_1 \ldots c_m$ such that each pixel x_i is constrained to be at most 1 greater or lesser than its neighbors (while enforcing positivity).

The difference of 1 is drawing attention to the fact that we have an ordinal relationship e.g. that the central pixel is larger than one neighbor. For a neighborhood $y_1 \ldots y_n$ with central pixel y_c the weakest way we can interpret this circumstance (assuming an image is encoded using integers) is that $y_p - y_c \leq 1$ where y_p is the larger pixel.

For a single 3x3 region and its LBP coding (its glyph) we have 8 of these kinds of relations. And, of course we know all pixels $x_1 \ldots x_n \geq 0$. See Figure 8 as an example of how we turn a glyph in to a set of inequality relations.

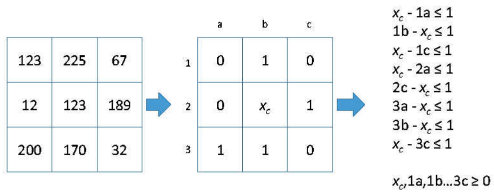

Figure 8. *Transformation of an LBP glyph into its associated constraints.*

Now we must consider how to recover an image given these inequalities. Our key insight is to employ a method called Quadratic Programming (QP). Per pixel we have 8 inequality relations where each pixel has a value between 0 and 255 inclusive. Let x_i denote the ith pixel in an image x (understanding that the pixel is in a a 2D grid, but for our purposes it is useful to think of there being n pixels in an image and x_i is the ith one). Now for neighboring pixels we have the linear inequalities as per Figure 8. But, thinking of the image as a vector of pixels these inequalities now refer to the ith and jth pixels where (i and j will be far apart).

We construct a large matrix A such that in each row we encode a single inequality relationship. So, in terms of our example the kth row at position i of A could have 1 and at position j a 0 if $x_i > x_j$. We have a corresponding vector B (which is a vector of 1s). Now we can write

$$Ax \geq B \tag{3}$$

In the above equation we are explicitly writing in matrix form all the inequalities that arise from every LBP glyph. Now we minimize:

$$min||x||s.t.Ax \geq 1 \tag{4}$$

We solve for the above using QP. QP is guaranteed to find the global optimum solution.

For an LBP glyph there are 8 comparisons between a central pixel and its neighbors. As such there are 8 constraints defined per pixel. Each pixel appears in 9 different neighborhoods so the transitive relationships between neighborhoods in an image will naturally be preserved.

Intuitively we can apply the same methodology to the Sudoku pattern. But, now there will be more constraints. Indeed in a 3x3 neighborhood the total number of binary comparisons possible is $\binom{9}{2} = 36$. Forming these as above gives us an exact representation of the Sudoku encoding and solving with QP remains the same.

In Figure 9 we show examples of Quadratic Reconstruction (QR) using LBP and Sudoku information on MATLAB's "cameraman" image.

Tone curve mapping

As a further processing stage in our QR pipeline we perform tone mapping on the calculated images using Isotonic Regression. This finds a least squares fit y to a vector x based on a known quantity x'. This is subject to the constraints $y_i \leq y_{i+1}$ [15]. This enforced monotonicity retains the illumination invariant properties of LBP and Sudoku. By finding a least squares fit between the original image and the new image we bring the QR pixel intensities more in line with that of the original image. For examples see Figure 10.

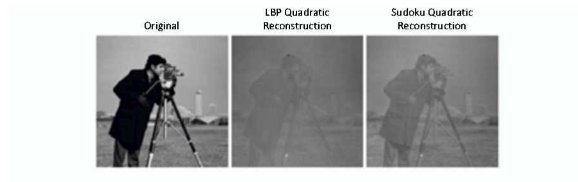

Figure 9. *Examples of quadratic reconstruction on "Cameraman.tif".*

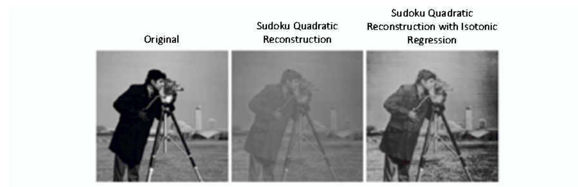

Figure 10. *Example of Isotonic Regression on a Sudoku Quadratic Reconstruction of MATLABs "Cameraman.tif".*

Of course, in the texture representation we do not know the original image so cannot really carry out Isotonic Regression. But, plausibly a tone curve could be stored along with texture glyphs which could be deployed if a reconstruction was necessary.

Finally, all the discussion to this point has been for grayscale images. For the purposes of this paper we assume we have texture glyphs (Sudoku or LBP) calculated for all 3 channels of an RGB image. The color reconstruction problem then solves the QP optimization (or minimum contrast) once per color channel.

Experiments

For all of our experiments we employ the pipeline detailed in Figure 3. To compare our reconstructions with the original image we use the Structural Similarity index (SSIM). This is a perceptual measure based on image degradation [16, 17]. We calculate the SSIM in each color channel and then take the mean of the three results. For QR experiments Isotonic Regression is performed on each individual channel. MC results are rescaled to be between 0 and 255.

We reconstruct 6 images in 6 different ways:

- LBP Minimum Contrast (LMC)
- Sudoku Minimum Contrast (SMC)
- LBP Quadratic Reconstruction (LQR)
- Sudoku Quadratic Reconstruction (SQR)
- LBP Quadratic Reconstruction with Isotonic Regression (LQRI)
- Sudoku Quadratic Reconstruction with Isotonic Regression (SQRI)

We use 3 images from the MATLAB default package: kobi.png, onion.png and football.jpg. We also use 3 images from the Outex_TC_00013 texture dataset: 000087.bmp, 000366.bmp and 000397.bmp. Table 1 shows our results. Figure 11 shows the images corresponding to Table 1.

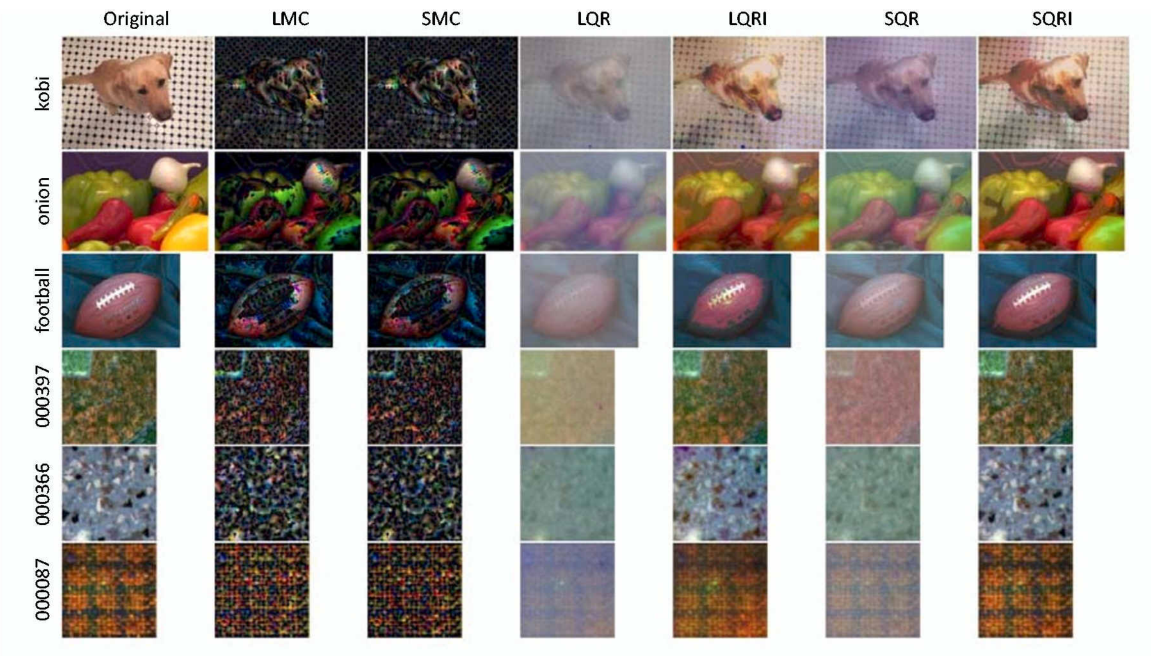

Figure 11. *All resultant images from our experiments.*

	LMC	SMC	LQR	LQRI	SQR	SQRI
kobi	0.20	0.20	0.29	0.53	0.46	**0.64**
onion	0.37	0.35	0.50	0.56	0.67	**0.71**
football	0.30	0.28	0.53	0.59	0.75	**0.82**
000397	0.32	0.30	0.64	0.89	0.80	**0.98**
000366	0.28	0.27	0.75	0.89	0.90	**0.96**
000087	0.40	0.40	0.49	0.70	0.63	**0.97**

Table 1: SSIM results for the 6 methods over the 6 images.

QR performs better than MC in all cases. Sudoku features also provide stronger reconstruction in QR but not MC. This is likely due to the shortcoming of Sudoku in MC, that is the reconstructed images codes only match closely, not exactly. This is due to how edges are computed: the maximum rank on an edge pixel can only be six and on a corner can only be four. This leads to a loss of information across the entire boundary of the image which then propagates into all other neighborhoods. It might be that the MC reconstructions could be improved in implementation however this is not within the scope of this paper.

Using Isotonic Regression to tone map our images also significantly increases performance across all six images. From a color perspective it is clear that the reconstructions using tone mapping are much closer to the originals however visually we believe there is still room for improvement. Isotonic Regression forms large areas of uniform intensity in the regular images which effects a loss of detail. Other methods of tone mapping could be employed to alleviate this, for example Histogram Mapping [18].

QR performs significantly better on the texture images than on the regular images. This is due to the fact that the texture images selected contain fine detail, or high frequency, patterns. As we only sample neighboring pixels it is expected that the high frequency information propagates more efficiently than low frequency. A possible solution for regular images would be to combine multiple different scales in the solution.

Conclusions

We have presented a novel use of Quadratic Programming applied to image recovery. We have shown that our QR method provides a statistically more similar image using a perceptual metric.

QR does have drawbacks - it is currently very computationally expensive. There are methods of circumventing this such as sparser sampling of the input image and this will be explored in the future. QP itself is the expensive part of the process and this is the primary source of slowdown.

Perceptual comparison is also interesting. Isotonic Regression provides a mathematically more similar image in all cases. However for the regular images we believe that the reconstruction without Isotonic Regression is more perceptually similar. A more complete set of results would include perceptual testing however this is not within the scope of this paper.

In conclusion we have shown that given an image's textural information it is possible to obtain a recognizable image. We have also shown that with our method computed in the *RGB* channels the recovered image can have a good approximation of the color of the original.

References

[1] R. M. Haralick, K. Shanmugam and I. Dinstein. Textural features for image classification. IEEE Transactions on Systems, Man and Cybernetics, IEEE, 6, pgs. 610-621. (1973).

[2] N. Jhanwar et al. Content based image retrieval using motif cooccurrence matrix. Image and Vision Computing, 22(14), pgs. 1211-1220. (2004).
[3] V. Arvis et al. Generalization of the cooccurrence matrix for colour images: application to colour texture classification. Image Analysis & Stereology, 23(1), pgs 63-72. (2011).
[4] D. Gabor, Theory of communication. Journal of the Institute of Electrical Engineers, 93, pgs 429457. (1946).
[5] A. Kumar and G. K. H. Pang, Defect detection in textured materials using Gabor filters. IEEE Transactions on Industry Applications, IEEE, 38(2), pgs.425-440. (2002).
[6] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor Filters. Pattern Recognition, 24(12), pgs. 1167-1186. (1991).
[7] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. Proc. of the Seventh IEEE International Conference on Computer Vision, IEEE, vol. 2, pgs. 1033-1038. (1999).
[8] H.J. Nussbaumer. Fast Fourier transform and convolution algorithms. Springer Science and Business Media, vol. 2. (2012).
[9] T. Ojala et al. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), pgs.971-987. (2002).
[10] S. Brahnam et al. Local binary patterns: new variants and applications. Springer Berlin Heidelberg. 2014.
[11] G. Finlayson and S. Nixon. Sudoku texture classification. Electronic Imaging, pgs. 1-5. (2016).
[12] B. M. Waller et al. Image reconstruction from local binary patterns. IEEE International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), IEEE, pgs. 118-123. (2013).
[13] T. Ahonen et al. "Face recognition with local binary patterns." European Conference on computer vision. Springer, Berlin and Heidelberg, pgs. 469-481. (2004).
[14] S.M. Lee et al. Evaluation of image similarity by histogram intersection. Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Franais de la Couleur.30(4), pgs 265-274.(2005).
[15] R. E. Barlow. Statistical inference under order restrictions; the theory and application of isotonic regression. Wiley-Blackwell, Columbia, MO. 1972.
[16] Z. Wang et al. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, IEEE, 13(4), pgs. 600-612. (2004).
[17] Z. Wang et al. Multiscale structural similarity for image quality assessment. Signals, Systems and Computers, pgs. 1398-1402. (2004).
[18] G. Finlayson et al. Illuminant and device invariant colour using histogram equalisation. Pattern Recognition. 38(2), pgs. 179-190. (2005).

Author Biography

Seth Nixon is a PhD student in the School of Computing Sciences at the University of East Anglia. He also conducted his undergraduate studies there and obtained a BSc in Computer Science. His work is primarily focused in the areas of color and texture analysis.

Graham Finlayson is a professor in the School of Computing Sciences at the University of East Anglia where he leads the world renowned Colour Lab. His interests span the development of physics-based image processing algorithms, to their implementation in embedded devices and, ultimately, to their commercialisation. Colour Lab technology ships in 100s of millions of devices. Professor Finlayson serves as a fellow of the Institute for Engineering Technology, the Royal Photographic Society and the Society for Imaging Science and Technology.