Texture Exemplars for Defect Detection on Random Textures

Xianghua Xie and Majid Mirmehdi

Department of Computer Science,
University of Bristol, Bristol BS8 1UB, England
{xie, majid}@cs.bris.ac.uk

Abstract. We present a new approach to detecting defects in random textures which requires only very few defect free samples for unsupervised training. Each product image is divided into overlapping patches of various sizes. Then, density mixture models are applied to reduce groupings of patches to a number of textural exemplars, referred to here as texems, characterising the means and covariances of whole sets of image patches. The texems can be viewed as implicit representations of textural primitives. A multiscale approach is used to save computational costs. Finally, we perform novelty detection by applying the lower bound of normal samples likelihoods on the multiscale defect map of an image to localise defects.

1 Introduction

Visual inspection has been one of the major applications of computer vision since the early 1980s. Numerous works have reported on detecting imperfections on a variety of surfaces [1,2,3], such as textile, ceramics, and wood. Some of the materials display complex patterns but appear visually regular on a larger scale, e.g. textile. Some others, such as printed ceramic tiles, may display very complex patterns that are random in appearance. Detecting subtle local defects on such surfaces turns out to be rather difficult [3].

A variety of statistical techniques have been investigated for defect detection, such as graylevel co-occurrence matrices. For those materials that exhibit a high degree of regularity and periodicity, e.g. textiles, template-based methods and Fourier-domain analysis have also proved useful for defect detection. Amongst other filtering-based techniques, Gabor filters have been applied, as shown in [2], due to their ability to analyse texture by achieving optimal joint localisation in the spatial and frequency domains. Randen and Husøy [4] present a thorough comparative review of texture analysis using filtering techniques.

However, the supremacy of filter bank based methods have been challenged by several authors. For instance, in [5], Varma and Zisserman argued that a large variety of signals (e.g. textures) can be analysed by just looking at small neighbourhoods. They used 7×7 patches to generate a texton based representation and achieved better performance than the filtering based methods they compared against when classifying material images from the Columbia-Utrecht

S. Singh et al. (Eds.): ICAPR 2005, LNCS 3687, pp. 404–413, 2005.

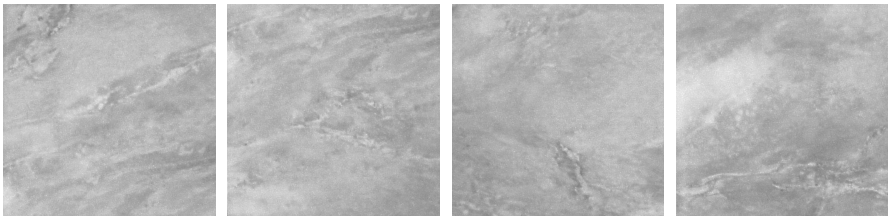

Fig. 1. Example marble tiles from the same family whose patterns are different but visually consistent

database. The results demonstrated that textures with global structures can be discriminated by examining the distribution of local measurements. This is a key factor in our approach in this paper. In [6], the authors also advocated the use of local pixel neighbourhood processing in the shape of local binary patterns as texture descriptors. Other works based on local pixel neighbourhoods are those that apply Markov Random Field (MRF) models, e.g. [1,7], where the inspection process was treated as a hypothesis testing problem on the statistics derived Recently in [8], Jojic et al. defined the epitome as a miniature, condensed version of an image containing the constitutive elements of its shape and textural properties needed to reconstruct the image. The epitome also relies on *raw* pixel values to characterise textural and colour properties rather than popular filtering responses. An image is defined by its epitome and a smooth, *hidden* mapping from the epitome to image pixels.

Inspired by the success of non-filtering local neighbourhood approaches, in this paper we propose a new approach to detecting and localising defects on random (or regular) textured surfaces. In a random texture application such as ceramic tile production, the images may appear different in textural pattern from one to another. However, the visual impression of the same product line remains consistent, e.g. see Figure 1. There exist textural primitives that impose consistency within the product line. Instead of recovering all the variations amongst images from a relatively large number of samples in a supervised manner for a traditional classification approach [9], we learn, in unsupervised fashion, textural primitive information from a very small number of training samples. We name these representations *texture exemplars* or *texems*.

Recently, the authors in [10] proposed novelty detection for classification of tiles using eigenfilters, but were not able to localise defects, essential when it is necessary to understand the nature and formation of the defects. Novelty detection is important from a practical viewpoint, not only because it is difficult to collect a wide range of defective samples for training for a more traditional classification based approach, such as neural networks, but also because some defects are usually unpredictable and occur only during production. To ensure computational efficiency we also extend the overall method into a multiscale framework.

In section 2, the proposed method is presented, including learning the texems, the multiscale approach, and the novelty detection stage. Experimental results are given in section 3. Section 4 concludes the paper.

2 Proposed Method

We consider that each product image is produced by putting together a certain number of subimage patches of various sizes, possibly overlapped. As the images of the same product contain the same textural elements, one product image can be generated from the patches extracted from other images. Thus, for a few given samples we can easily obtain a large number of patches of various sizes (which can in turn generate a large set of new images with the same visual impression). However, it is computationally prohibitive to perform defect detection based on such a large number of patches. Also, the patches themselves contain lots of redundant information. We can reduce the number of patches by learning a relatively small number of primitive representatives, i.e. texems.

The proposed method is related to the texton model in the sense that both try to characterise textural images by using micro-structures. Textons were first introduced by Julesz [11] as the atoms of pre-attentive human visual perception. An image is considered as a superposition of a number of image bases selected from an over-complete dictionary. The image bases are generated by a smaller number of texton elements, selected from a dictionary of textons [12]. Textons have attracted much attention in vision applications, including image classification and motion modelling. Recently, in [12], Zhu et al. presented generative models for learning the fundamental image structures from textural images. However, the proposed method is significantly different from the texton model in that it relies directly on the subimages instead of using base functions. The texems are implicit representations of textural primitives, which makes them more flexible as they come at different sizes, while textons are explicit representations. For example, if the texem size reduces to a single pixel, it becomes histogram analysis. If the texems are the same size as the input images, then the problem turns into image template analysis. Each texem indeed becomes a template. In general our texems contain multiple textural primitives which as a whole describe a family of textures. This implicit representation at various sizes avoids the difficulties of explicitly finding the best primitive representation, e.g. the optimum window size as in the case of textons (for example see [5]).

In brief, we break down a defect free image into overlapping paches of various sizes, and group similar sized patches into a multidimensional space, dependent on the patch size, and describe the clusters found using a Gaussian mixture model. The representative texture exemplars are then learned through an EM algorithm applied on the mixture density parameters. Then, as we are interested in localising the defective regions, we extract a small patch at each pixel position of the testing image and classify it using the set of texems obtained at the training stage.

2.1 Learning Textural Exemplars (Texems)

The texture exemplars, referred to as *texems*, are image representations at various sizes that encapsulate the texture or visual primitives of a given image. For instance, in the case of an example random texture, the textural primitives are

consistent from one image to another, hence texems can characterise a family of images of the random texture. Each texem, denoted as $\mathbf{m}$, is defined by a mean, $\boldsymbol{\mu}$, and a corresponding covariance matrix, $\boldsymbol{\omega}$, i.e. $\mathbf{m} = \{\boldsymbol{\mu}, \boldsymbol{\omega}\}$.

The original image $\mathbf{I}$ is broken down into a set of P patches $\mathbf{Z} = \{\mathbf{Z}_i\}_{i=1}^{P}$, each containing pixels from a subset of image coordinates. The shape of the patches can be arbitrary, but in this study we used square patches of size $d = N \times N$. The patches may overlap and can be of various sizes, e.g. as small as 5×5 to as large as required (here 20×20). We assume that there exist K texems, $\boldsymbol{\mathcal{M}} = \{\mathbf{m}_k\}_{k=1}^{K}$, $K \ll P$, for image $\mathbf{I}$ such that each patch in $\mathbf{Z}$ can be generated from a texem with certain added variations. In other words, the original image $\mathbf{I}$ can be reconstructed by the texems with a certain reconstruction error.

To learn these texems the P patches are projected into a set of higher dimensional spaces. The number of these spaces is determined by the number of different patch sizes and their dimensions are defined by the corresponding value of d. Each pixel position contributes one coordinate of a space. Each point in a space corresponds to a patch in $\mathbf{Z}$. Then each texem and its covariance matrix represent a class of patches in the corresponding space. We assume that each class is a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\omega}_k$, which corresponds to $\mathbf{m}_k$ in the spatial domain. Thus, the probability density function for a particular patch $\mathbf{Z}_i$ given that it belongs to the kth texem $\mathbf{m}_k$, is:

$$p(\mathbf{Z}_i|\mathbf{m}_k, \theta) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\omega}_k|}} \exp\{-\frac{1}{2}(\mathbf{Z}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\omega}_k^{-1} (\mathbf{Z}_i - \boldsymbol{\mu}_k)\}, \tag{1}$$

where $\theta = \{\boldsymbol{\alpha}_k, \boldsymbol{\mu}_k, \boldsymbol{\omega}_k\}_{k=1}^{K}$ is the parameter set containing $\boldsymbol{\alpha}_k$, which is the *prior* probability of kth texem constrained by $\sum_{k=1}^{K} \boldsymbol{\alpha}_k = 1$, the mean $\boldsymbol{\mu}_k$, the covariance $\boldsymbol{\omega}_k$. Since all the texems $\mathbf{m}_k$ are unknown, the parameter set θ can be determined first by marginalizing the joint distribution by summing across the texems, $p(\mathbf{Z}_i|\theta)$, and then optimising the data log-likelihood expression of the entire set $\mathbf{Z}$, given by

$$\log p(\mathbf{Z}|K, \theta) = \Sigma_{i=1}^{P} \log p(\mathbf{Z}_i|\theta) = \Sigma_{i=1}^{P} \log(\Sigma_{k=1}^{K} p(\mathbf{Z}_i|\mathbf{m}_k, \theta)\boldsymbol{\alpha}_k). \tag{2}$$

Hence, the objective is to estimate the parameter θ for a given number of texems. The *Expectation Maximization* (EM) technique can be used to find the maximum likelihood estimate of our mixture density parameters from the given data set $\mathbf{Z}$. That is to find $\hat{\theta}$ where

$$\hat{\theta} = \arg\max \log(\mathcal{L}(\theta|\mathbf{Z})) = \arg\max \log p(\mathbf{Z}|K, \theta). \tag{3}$$

Then the two steps of the EM stage are as follows. The E-step involves a soft-assignment of each patch $\mathbf{Z}_i$ to texems, $\boldsymbol{\mathcal{M}}$, with an initial guess of the true parameters, θ. We denote the intermediate parameters as $\theta^{(t)}$. The probability that patch $\mathbf{Z}_i$ belongs to the kth texem may then be computed using Bayes rule:

$$p(\mathbf{m}_k|\mathbf{Z}_i, \theta^{(t)}) = \frac{p(\mathbf{Z}_i|\mathbf{m}_k, \theta^{(t)})\boldsymbol{\alpha}_k}{\Sigma_{k=1}^{K} p(\mathbf{Z}_i|\mathbf{m}_k, \theta^{(t)})\boldsymbol{\alpha}_k}. \tag{4}$$

The M-step then updates the parameters by maximizing the log-likelihood, resulting in new estimates:

$$\hat{\boldsymbol{\alpha}_k} = \frac{1}{P}\Sigma_{i=1}^{P} p(\mathbf{m}_k|\mathbf{Z}_i, \theta^{(t)}), \tag{5}$$

$$\hat{\boldsymbol{\mu}_k} = \frac{\Sigma_{i=1}^{P} \mathbf{Z}_i p(\mathbf{m}_k|\mathbf{Z}_i, \theta^{(t)})}{\Sigma_{i=1}^{P} p(\mathbf{m}_k|\mathbf{Z}_i, \theta^{(t)})},$$

$$\hat{\boldsymbol{\omega}_k} = \frac{\Sigma_{i=1}^{P} (\mathbf{Z}_i - \hat{\boldsymbol{\mu}_k})(\mathbf{Z}_i - \hat{\boldsymbol{\mu}_k})^T p(\mathbf{m}_k|\mathbf{Z}_i, \theta^{(t)})}{\Sigma_{i=1}^{P} p(\mathbf{m}_k|\mathbf{Z}_i, \theta^{(t)})}.$$

The E-step and M-step are iterated until the estimations are stabilises. Then, the texems can be easily obtained by projecting the parameters back to the spatial domain. Various sizes of texems can be used and they can overlap to ensure they capture sufficient textural characteristics.

2.2 A Simple Multiscale Approach

In order to capture sufficient textural properties, texems can be from as small as 3×3 to larger sizes such as 20×20. However, the dimension of the space we transform patches $\mathbf{Z}$ into will increase dramatically as the dimension of the patch size d increases. This means that a very large number of samples and high computational costs are needed in order to accurately estimate the pdf in very high dimensional spaces, forcing the procurement of a large number of training samples. Therefore, instead of generating variable-size texems, we learn fixed size texems in a multiscale. This will result in (multiscale) texems with a very small size, e.g. 5×5. A simple multiscale approach by using a Gaussian pyramid is sufficient.

Let us denote $\mathbf{I}^{(n)}$ as the nth level image of the pyramid, $\mathbf{Z}^{(n)}$ as all the image patches extracted from $\mathbf{I}^{(n)}$, l as the total number of levels, and $S^{\downarrow}$ as the down-sampling operator. We then have $\mathbf{I}^{(n+1)} = S^{\downarrow} G_{\sigma}(\mathbf{I}^{(n)})$, $\forall n, n = 1, 2, ..., l-1$, where G_{σ} denotes the Gaussian convolution. The finest scale layer is the original image, $\mathbf{I}^{(1)} = \mathbf{I}$. We then extract multiscale texems from the image pyramid using the method presented in the previous section. Similarly, let $\mathbf{m}^{(n)}$ denote the nth level of multiscale texems and $\theta^{(n)}$ the parameters associated at the same level, which will then be used for novelty detection at the corresponding level of the pyramid. During the EM process, the stabilised estimation of a coarser level is used as the initial estimation for the finer level, i.e. $\hat{\theta}^{(n,t=0)} = \theta^{(n+1)}$, which helps speed up the convergence and achieve a more accurate estimation.

2.3 Novelty Detection

Once the texems are obtained from a single training image, we then can work out the minimum bound of normal samples in each resolution level in order to perform novelty detection. A small set of defect free samples (e.g. 4 or 5 only) are arranged within a multiscale framework, and patches with the same texem size are extracted. The probability of a patch $\mathbf{Z}_i^{(n)}$ belonging to texems in

the corresponding nth scale is $p(\mathbf{Z}_i^{(n)}|\theta^{(n)}) = \Sigma_{k=1}^{K} p(\mathbf{Z}_i^{(n)}|\mathbf{m}_k^{(n)}, \theta^{(n)})\boldsymbol{\alpha}_k^{(n)}$. The minimum probability of a patch $\mathbf{Z}_i^{(n)}$ at level n across the training images is treated as the lower bound of the data likelihood, denoted as $\Lambda^{(n)}$:

$$\Lambda^{(n)} = \min(p(\mathbf{Z}_i^{(n)}|\theta^{(n)})), \qquad \forall\ \mathbf{Z}_i^{(n)} \in \mathbf{Z}^{(n)}. \tag{6}$$

This completes the training stage in which with only a very few non-defective images, we determine the texems and an automatic threshold for marking new image patches as good or defective.

In the testing stage, the image under inspection is again layered into a multiscale framework and patches at each pixel position (x, y) at each level n are examined against the learned texems. The probability for each patch is then calculated, $p(\mathbf{Z}_i^{(n)}|\theta^{(n)})$, and compared to the minimum data likelihood, $\Lambda^{(n)}$, at the corresponding level. Let $Q^{(n)}(x, y)$ be the probability map at the nth resolution level. Then, the potential defect map, $\mathcal{D}^{(n)}(x, y)$, at level n is:

$$\mathcal{D}^{(n)}(x, y) = \begin{cases} 0 & \text{if } Q^{(n)}(x, y) \geq \Lambda^{(n)} \\ \Lambda^{(n)} - Q^{(n)}(x, y) & \text{otherwise.} \end{cases} \tag{7}$$

We then need to combine the information coming from all the resolution levels to build the certainty of the defect at position (x, y). We follow a method described in [2] which combines information from different levels of a multiscale pyramid and reduces false alarms. It assumes that a defect must appear in at least two adjacent resolution levels for it to be certified as such. Using a logical AND, implemented through the geometric mean, of every pair of adjacent levels, we initially obtain a set of combined maps as:

$$\mathcal{D}^{(n,n+1)}(x, y) = [\mathcal{D}^{(n)}(x, y)\mathcal{D}^{(n+1)}(x, y)]^{1/2}. \tag{8}$$

Please note that each $\mathcal{D}^{(n+1)}(x, y)$ is scaled up to be the same size as $\mathcal{D}^{(n)}(x, y)$. This operation reduces false alarms and yet preserves most of the defective areas. Next, the resulting $\mathcal{D}^{(1,2)}(x, y)$, $\mathcal{D}^{(2,3)}(x, y)$, ..., $\mathcal{D}^{(l-1,l)}(x, y)$ are combined in a logical OR, as the arithmetic mean, to provide a final map for the defects detected across all the scales:

$$\mathcal{D}(x, y) = \frac{1}{l-1} \sum_{n=1}^{l-1} \mathcal{D}^{(n,n+1)}(x, y), \tag{9}$$

where $\mathcal{D}(x, y)$ contains the joint contribution of all the resolution scales and marks the defects.

3 Experimental Results

We applied the proposed method to a variety of tile data sets with different types of defects including physical damage, pin holes, textural imperfections, pattern mis-registrations, and many more. The test samples, at 512×512 pixels,

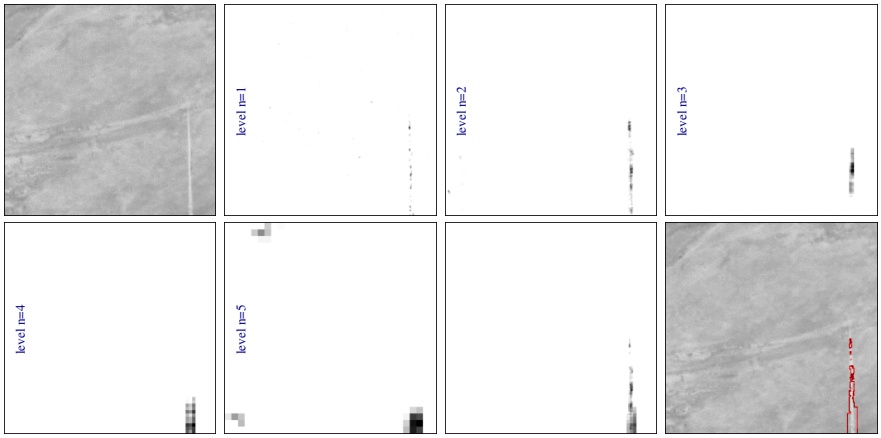

Fig. 2. Localising textural defects - from top left to bottom right: original defective tile image, detected defective regions at different levels $n = 1, 2, ..., 5$, the joint contribution of all resolution levels, and the final defective regions superimposed on the original image.

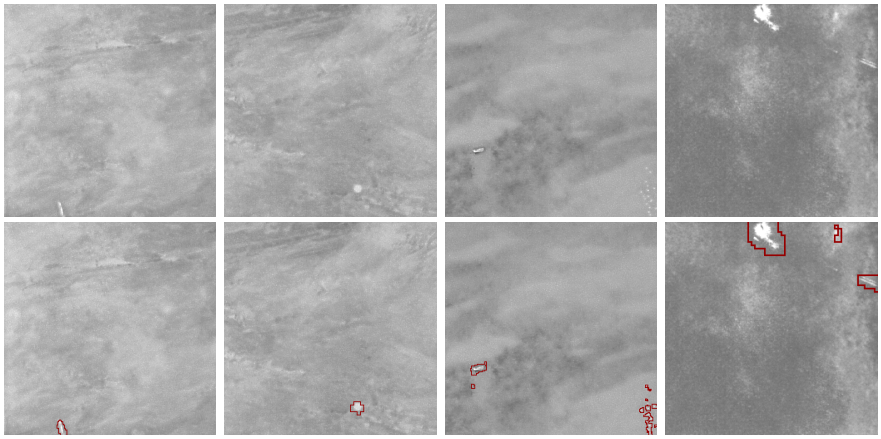

Fig. 3. Defect detection - first row: original images, second row: superimposed defective regions from left - surface defect, small bump, surface defect and a cluster of pin holes, and missing print.

were appropriately pre-processed to assure homogeneous luminance, spatially and temporally. In our experiments, only one defect free sample was used to extract the texems, and only five to generate the lower bound data likelihoods $\Lambda^{(n)}$. The number of texems at each level were empirically set to 12, and the size of each texem was set to 5×5 pixels. The number of multiscale levels was $l = 5$. These parameters were fixed throughout our experiments on a variety of random texture tile prints.

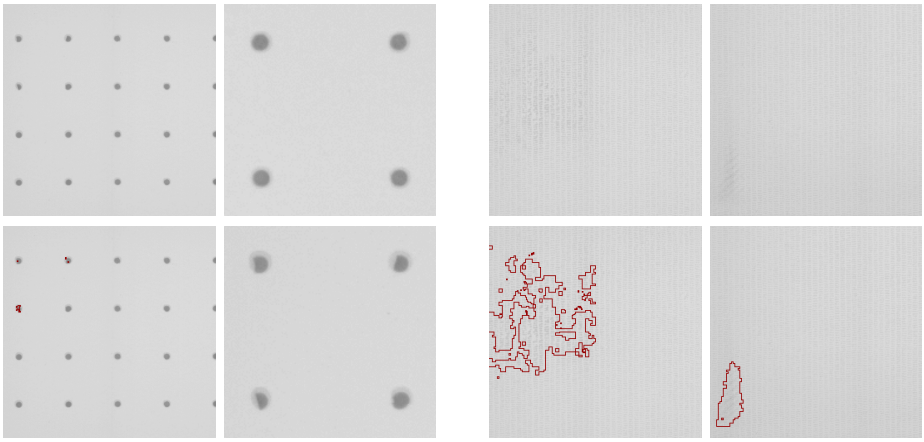

Fig. 4. Detecting defects in regular patterns - first column: original defective image and superimposed defective regions; second column: closeup views of normal and abnormal dot patterns from the previous image; third and fourth columns: two defective samples of a different regular texture with pattern irregularities and superimposed defective areas.

Figure 2 shows a random texture example, from the same family as in Figure 1, with a defect in the lower right region introduced by a printing problem. The detected potential defective regions at each resolution level n, $n = 1, 2, ..., 5$ are marked on the corresponding images in Figure 2. It can be seen that the texems show good sensitivity to the defective region at different scales. As the resolution progresses from coarse to fine, additional evidence for the defective region is gathered. This evidence is then combined, shown in the bottom-right of Figure 2, to produce the defect map $\mathcal{D}$. The final image shows the superimposed defects on the original image. As mentioned earlier, the defect fusion process can eliminate false alarms, e.g. see the extraneous false defect regions in level $n = 5$ which disappear after the operations in (8) and (9).

More examples of different textures are shown in Figure 3. In each family of patterns, the textures are varying but of the same visual impression. In each case the proposed method could find from very small surface defects to large variable shaped defects such as the missing print as shown in the last example.

The proposed method can also detect defects in regular patterns. For example, the first two images of Figure 4 show three incompletely printed dots at the top-left corner of the regular pattern. Each dot is composed of one larger, lighter dot as background and one smaller, darker dot positioned in the centre (see the closeup view in the second column of the Figure 4). In other two examples in Figure 4, printing error and smudge defects damaging the local pattern regularity in a grid-like pattern were correctly detected.

Next, we compare our results with those obtained by using the epitome [8]. Two example cases are shown in Figure 5. We apply the epitome for texture segmentation (with software provided by the authors of [8]), however, we

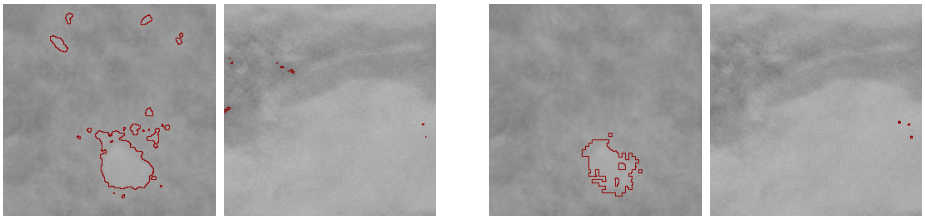

Fig. 5. Leftmost two images: examples of novelty detection using epitomes which produced many false positives or failed to locate the true defects. Rightmost two: novelty detection results using texems which successfully detected all the true defects (a print defect and three pin holes).

extend it into a similar framework as the proposed method for better comparative analysis. Hence, we generate the multiscale version of an image and at each scale, we learn an appearance epitome using 5×5 image patches at each scale, resulting in epitomes varying from 7×7 to 24×24. Using these multiscale epitomes, we perform novelty detection, similar to the method described in Section 2.3. It involves finding a match in the epitome for an image patch under inspection. As the epitome is still larger than the patch itself and there are numerous comparisons across the image, the detection procedure is computationally very expensive. The results show that our method is less sensitive to false alarms.

As patches are extracted from each pixel position at each resolution level, a typical *training stage* involves examining over 0.25 million patches (for a 512×512 image) to learn the texems in multiscale. This takes around 25 minutes on an AMD Athlon XP Processor (1.4GHz) to obtain the texems and to determine the thresholds for novelty detection. The testing stage is much faster, requiring about one minute to inspect one tile image. However, it will cost the epitome based method several hours to perform training or testing. The computation time of our method can be greatly reduced by examining every other pixel (or fewer).

The examples show the ability of texems ability in localising small or large defects on highly textured surfaces. We evaluated our defect detection rate across 1512 tiles from eight different families of textures and obtained very good results with 95.87% sensitivity, 89.47% specificity, and 92.67% overall accuracy.

4 Conclusions

We presented an automatic defect detection and localisation algorithm for random textures. The proposed method only trained on a very small number of defect free samples with the aid of novel texems that are implicit representations of primitive textural information. The texems are at present only applicable to graylevel images and we intend to extend them to colour analysis. This can be achieved by modifying the inference procedure that derives them. The computational needs of the method are somewhat demanding for a real-time inspection.

We shall investigate various avenues to achieve a rate of around 2-4 surfaces per second which is an acceptable tile industry norm. While we present this work with respect to ceramic tiles, the proposed method should be suitable to other flat textured surfaces, such as textiles and wood.

Acknowledgments

This work is funded by EC project `G1RD-CT-2002-00783 MONOTONE`, and X. Xie is partly funded by the ORSAS, Universities UK.

References

1. Cohen, F., Fan, Z., Attali, S.: Automated inspection of textile fabrics using textural models. IEEE T-PAMI **13** (1991) 803–809
2. Escofet, J., Navarro, R., Millán, M., Pladellorens, J.: Detection of local defects in textile webs using Gabor filters. Opt. Eng. **37** (1998) 2297–2307
3. Kittler, J., Marik, R., Mirmehdi, M., Petrou, M., Song, J.: Detection of defects in colour texture surfaces. In: IAPR MVA. (1994) 558–567
4. Randen, T., Husøy, J.: Filtering for texture classification: a comparative study. T-PAMI **21** (1999) 291–310
5. Varma, M., Zisserman, A.: Texture classification: Are filter banks necessary? In: CVPR. (2003) 691–698
6. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE T-PAMI **24** (2002) 971–987
7. Özdemir, S., Baykut, A., Meylani, R., Ercil, A., Ertüzün, A.: Comparative evaluation of texture analysis algorithms for defect inspection of textile products. In: CVPR. (1998) 1738–1740
8. Jojic, N., Frey, B., Kannan, A.: Epitomic analysis of appearance and shape. In: ICCV. (2003) 34–42
9. Kumar, A.: Neural network based detection of local textile defects. PR **36** (2003) 1645–1659
10. Monadjemi, A., Mirmehdi, M., Thomas, B.: Restructured eigenfilter matching for novelty detection in random textures. In: BMVC. (2004) 637–646
11. Julesz, B.: Textons, the element of texture perception and their interactions. Nature **290** (1981) 91–97
12. Zhu, S., Guo, C., Wang, Y., Xu, Z.: What are textons? IJCV **62** (2005) 121–143