Chapter 5

Application to surface inspection problems

All the previous chapters define the necessary conditions to broach the problem of surface inspection, basically on industrial problems. This term includes a wide range of problems but we will centre in those where texture and colour simultaneously are the key factors in the problem solution. The work presented here began with an application devoted to the classification of coloured and textured samples of ceramic tiles. Nonetheless, the tools and methods used can be applied for many similar applications.
Considering the perceptual operator introduced in the previous chapter, here we propose a computational colour texture representation based on a multiscale approach. In it, colour measurements are done considering the perceptual blurring that simulates a large distance observer position, and the proposed sharpening to simulate a short distance observer position, from which the blob extraction is derived. From these blobs the features of textons associated to colours are defined.
Finally, we will apply this approach to the classification of ceramic tiles, and to the problem of printing quality classification.

5.1 Building a colour texture representation

In chapter 1 we have briefly introduced previous works on colour, texture and colour-texture computational representations. In chapter 4 we have explained induction phenomena of the human visual system that explains the interaction between these two visual cues when appear jointly on a surface.

Now we want to build a computational colour-texture representation that considers the induction phenomena. To this end, we will follow a common approach in computer vision that is building a feature vector that combines different image properties, but in this case we will make it to take into account the induction phenomena that are involved in the human perception of colour textures.

In figure 4.4 we have plot a model proposal to integrate the most common effects

of colour induction: assimilation and contrast. We have seen that the first can be computationally simulated by a perceptual blurring and the second can be computationally simulated by a perceptual sharpening. Their activity is complementary, the first one is produced when the spatial frequency is high and the second one is produced when the spatial frequency is low. This scale-dependent mechanisms are very common in multiscale approaches in computer vision, and it allows to extract different information of a given image as it is implementing a vision process of looking at the image from different observer positions or looking at different image regions in a more attentive process.

The colour assimilation allows to take a global colour measurement of an image region when observed from a long distance position and when blob details are lost. The colour contrast phenomena allows to take local measurements of the image blobs when observed separately, that is, from a short distance position. We have seen in chapter 1 the need for these two types of measurements, when we propose a colour-texture representation. In this case, and to consider colour and texture interactions we will define the measurement on a set of preprocessed images that will simulate the induction phenomena. These sets of images are constructed in the following steps:

Step 1 A given input image, I, has to be transformed to its opponent colour representation, as it has been seen in chapter 4, we will represent as $Opp(I)_i$ the i channel of the opponent representation of I.

Step 2 An assimilation process is simulated by building the following set of images:

$$\{A(Opp(I)_i, s)\} \tag{5.1}$$

where i represents the image channel and s represents the scale of the perceptual blurring, $A(I, s)$, that represents the convolution of the image I with a kernel defined by the s parameter. In this case, the convolution kernels can be built accordingly for different observer conditions, by considering the psychophysical measurements presented in the Spatial-CIELAB introduced in section 4.2.1. In this case the selected scale will be directed to tune the high spatial frequency image relationships, these set of images will be the basis for global measurements of the image colour.

Step 3 A chromatic contrast process is simulated by building the following set of images:

$$\{C(Opp(I)_i, s)\} \tag{5.2}$$

where i represents the image channel and s represents the scale of the perceptual sharpening, $C(I, s)$, which can be implemented by the spread sharpening defined in section 4.3.3. In this case we do not have the psychophysical measurements that can provide the needed parameters to represent the conditions for a given scale. Therefore we will only use this set of images for the blob segmentation step, but not for colour measurements.

Step 4 The results of the previous steps put the basis for a consistent blob segmentation step based on colour properties. The blob segmentation process will be explained in the following section. The set of k segments of a given sharpened image will be denoted as:

$$\{C^{sg}(I_i, s)\}_{sg:1\ldots k} \tag{5.3}$$

Once, the colour induction phenomena have been simulated on a set of blurred and sharpened images, now we want to measure colour and texture properties of these images.

In the next sections we will explain how to perform a blob segmentation based on colour properties, afterwards, we will define the global and local measurements that are usually computed for colour and texture representation, and finally we will build the complete computational colour-texture representation we want to propose.

5.2 Perceptual blob segmentation

In the representation of the colour texture that we propose, the blobs play an important role in the description and they should be treated carefully. In the next section we will describe the possible mechanism to obtain the set of images where which content are blobs of similar colour.

The first step to describe colour texture is to separate pixels on the image that share similar colour properties. This step can be seen as a colour-granulometric stage, where we sieve pixels of different colours. Given an image, the output of this segmentation step is a set of images containing the blobs of a specific colour. Although there are many ways to segment images most of them are based on its spatial contents (*region growing*, *split and merge*, etc) and they are not what we want. Our segmentation has to be independent of the spatial relations between colours. The objective is to return a set of sets of pixels where in each of these set the colour variance is minimum independently of the pixel location. From the segmentation methods we will focus on clustering methods, and from these ones on the bayesian approach.

We can do this as a supervised operation in which approximate colour coordinates are introduced to initialise each colour centre, or in unsupervised mode starting from algorithmically chosen centres. In the second case the clustering algorithm can have non–deterministic solutions, depending on the way these centres are chosen. However most of the times these algorithms tend to the same solution. This point should be kept in mind when working with colourful images.

The basic idea of cluster algorithms is that given an image $\mathbf{I}$ with c different colours, and given a set of initial colours $\{\mathbf{C}_1, \ldots, \mathbf{C}_p\}$ the segmentation step output is a set of images:

$$\{\mathbf{I}_1, \ldots, \mathbf{I}_c\}$$

where each image $\mathbf{I}_i$ contains all the blobs of a specific colour labeled i. Although there are many algorithms to cluster colour data, we have selected two of them for being widely used in computer vision. One of the methods is chosen for its simplicity

and the other one is based on the same idea but a more complex model is behind it. In the last case not only blobs are clustered but also it gives statistical information of its colour content that can be useful for posterior colour classification. These algorithms are the well known *k–means* and *expectation maximisation mixture of gaussians*. In both cases the number of clusters to obtain must be set a priori. To solve this problem a simple method is defined in [46] to analyse how many colour clusters appear in the image. In all of our cases the number of clusters is known since we have information of the production process or we know what we are looking for. We will summarise both clustering methods briefly that are described in detail in [26, 10].

The point of departure in any case should be an image where clustering is helped by its content i.e: the much separate the colours are the easier is the problem of clustering. If we manage colour separation by means of linear transform the problem remains the same as the proportions are maintained. So a non linear transform is needed. In chapter 4 we have introduced the colour contrast perceptual operators. In a normal situation they act to simulate human behaviour when looking at low frequencies patterns. In our examples we have high frequency texture, however we use the spread perceptual sharpening operator (Eq. 4.22) to simulate a human being looking at the sample at very high resolution: In this case colour contrast conditions operates and colour blobs are easier to segment. Figure 5.1 explains this effect where an original sample with their 2D–histogram distribution in the RGB and opponent space are depicted in the left column, and when applying the SS operator (right column) the colours in the image appear more clearly. We are not saying that colour contrast is the effect that helps humans to distinguish colour blobs, in fact if any process is done it is assimilation, but in order to better cluster colour data to describe how colour texture is, humans perform better at high resolution.

5.2.1 K-means clustering

Given a set of n–dimensional data, the goal of the k–means clustering is to find the centres of the colour cluster $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_c$ to be used as their prototypes, where c i the number of clusters that has been set a priori.

Starting form a set of initialised cluster prototypes $\Theta = \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ we want to know which is the probability for one pixel $\boldsymbol{x}_i$ to belong to the cluster ω_k, $P(\omega_k \mid \boldsymbol{x}_i, \Theta)$. The algorithm works with the assumption that if the distance between $\boldsymbol{x}_i$ and $\boldsymbol{\mu}_k$ is small the probability is large. Then, we compute the square Eculidean distance between the pixel and all the cluster prototypes, $\| \boldsymbol{x}_i - \boldsymbol{\mu}_p \|^2$. Let $\boldsymbol{\mu}_m$ be the nearest prototype to $\boldsymbol{x}_i$ then, we approximate $\hat{P}(\omega_k \mid \boldsymbol{x}_i, \Theta)$ as

$$\hat{P}(\omega_k \mid \boldsymbol{x}_i, \Theta) = \begin{cases} 1 & : \quad k = m \\ 0 & : \quad \text{otherwise.} \end{cases} \tag{5.4}$$

Following this schema each pixel of the image is set to belong to one cluster and from the pixels in each cluster the prototypes $\boldsymbol{\mu}_k$ are recalculated. This process is done until some stop condition is reached. This condition could be the number of iterations to do or the stability of prototypes of a minimum threshold to reach of the RMS error.

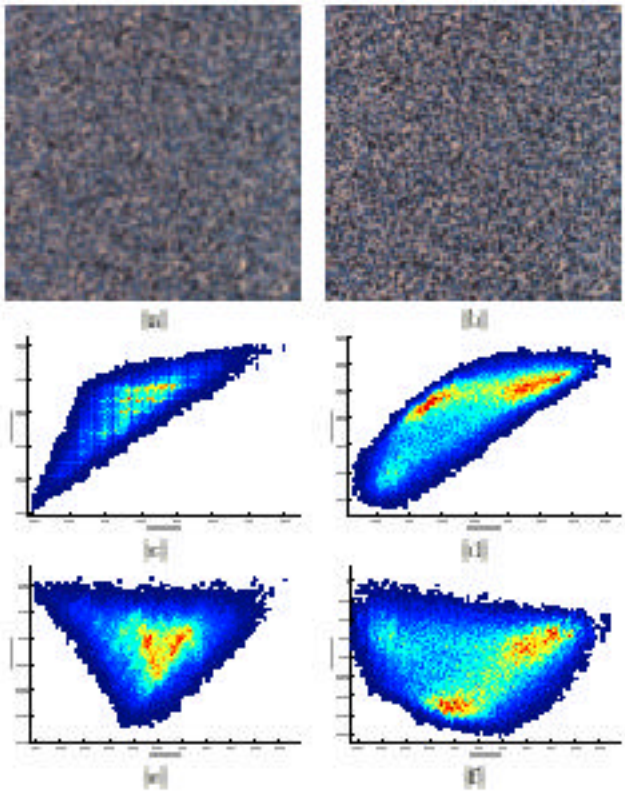


Figure 5.3: Colour distributions of two colour texture images (a) and (b) original and spatial perceptual discernment results. (c) and (d) the 2D-histogram relating one of the dimensions in the RGB space. (e) and (f) 2D-histogram in the opponent space.

Therefore, *k-Means* algorithm is performing a clustering step over the colour space, where clusters are formed considering the euclidean distance between colour coordinates of pixels in this space. Given that we are using an euclidean distance on a colour space we should do it on a perceptually uniform colour space as CIELAB or CIELUV. Some experiments have been done in this sense and they make us to conclude that, for the images of our applications, the segmented blobs are nearly the same, hence, there is no need to introduce a step that needs further calibration and does not improve the clustering result.

5.2.2 Parameter estimation of the colour distribution

In this section we assume the colour distribution as the sum of several gaussian distributions. To get an estimation of the parameters of these gaussian we will use an *Expectation–Maximisation* method.

This method assumes that data are generated from a set of gaussian distributions that when mixed form the final distribution. Its goal is to extract the parameters of the gaussians that best describes the distribution form the data themselves. It is done in a two steps process were in the first and starting from an initial guess an expectation of which are the best parameters is done. The second step takes the set of all possible parameter modifications and chooses the one that maximises the fitting of data with the mixture. With this new guess another iteration is done until some condition is reached (as in the case of k–means procedure). The degree of data fitting is measured in terms of likelihood $\mathcal{L} = \prod_{n=1}^{N} p(\boldsymbol{x}_n)$, where data are N points, $\boldsymbol{x}_n$, and $p(\boldsymbol{x}_n)$ is the a priori probability of a given point to happen. But to reduce the complexity of the problem the log–likelihood is used and then it becomes a minimisation problem of the expression

$$E = -\ln \mathcal{L} = -\sum_{n=1}^{N} \ln p(\boldsymbol{x}_n) \tag{5.5}$$

$p(\boldsymbol{x})$ is defined in term of the probability of belonging to the gaussian distribution of the mixture given a probability of each particular gaussian to occur, $p(\boldsymbol{x}) = \sum_{j=1}^{k} p((x) \mid j)P(j)$. In the case of mixture of gaussian and for the case of diagonal covariance to simplify the problem, the probability of a given d–dimensional point to belong to a certain gaussian distribution j is

$$p(\boldsymbol{x} \mid j) = \frac{1}{(2\pi)^{d/2}\sqrt{\mid \Sigma_j \mid}} e^{-\frac{(\boldsymbol{x}-\boldsymbol{\mu_j})'\Sigma_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu_j})}{2}}. \tag{5.6}$$

Then we can take derivatives on the unknown parameters of the gaussians maximise the likelihood. In this case $\frac{\partial E}{\partial \boldsymbol{\mu}_j}$ and $\frac{\partial E}{\partial \boldsymbol{\sigma}_j}$ for all j he details can be found in [10]. The a priori probability of a certain gaussian $P(j)$ has to be also derived to maximise $\mathcal{L}$. As our intention is jut to pose the method we will not focus in the mathematical description. From this maximisation step we obtain a set of parameters that define the configuration of the mixture, $\hat{\boldsymbol{\mu}}_j$, $\hat{\boldsymbol{\sigma}}_j$ and $\hat{P(j)}$ that are the initial guess for the next iteration.

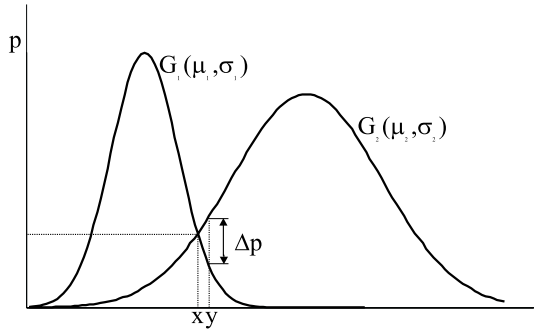


Figure 5.2: Decision criterion: when a point x can belong to more than one gaussian distribution it is rejected. In the example, y will be set as a point in G_2 because its differential to G_1 is large enough.

5.2.3 Decision criterion for clustering

When using mixture of gaussians the pixel $\boldsymbol{x}$ is classified as belonging to the cluster i with maximum posterior probability $p(i \mid \boldsymbol{x}, \Theta_i) = p(\boldsymbol{x}x \mid i, \Theta_i)P(i)/\sum_{j=1}^{c} p(\boldsymbol{x} \mid j, \Theta_j)P(j)$, where Θ_i are the parameters of the i–th gaussian. The original method does not take into account that some times a pixel can belong to two or more different gaussians with nearly equal probability. It is the case of point x in figure 5.2. Those pixels with this ambiguity should not be considered to avoid colours to shift from one cluster to another one. We have introduced a criterion to ensure that pixels are unambiguous. The idea is quite simple, instead of assigning a pixel to the class with the prior criterion, the posterior probabilities for each class are sorted and if the difference between the two maximum posteriors is greater than a certain percentage Δp then the pixel is assigned to the first cluster, in any other case the pixel is keep away in an *ambiguous cluster*. Then, $\mathcal{C}(\boldsymbol{x}, \boldsymbol{\Theta})$ as the expression to classify a pixel to a cluster i from a set of gaussian parameters $\boldsymbol{\Theta} = \Theta_1 \ldots \Theta_c$ is:

$$\mathcal{C}(\boldsymbol{x}, \boldsymbol{\Theta}) = \begin{cases} i & : \; p(i \mid \boldsymbol{x}, \Theta_i)(1 - \Delta p) \geq p(j \mid \boldsymbol{x}, \Theta_j) \quad \forall j = 1 \ldots c \wedge i \neq j \\ ambiguous & : \; \text{otherwise} \end{cases} \tag{5.7}$$

To finish this section we show an example of the two cluster techniques applied both to the input image an the perceptually sharpened image of figure 5.1(a) and (b). If the images are not reproduced with a high quality printing device the colour differences perhaps will not be seen and the attention should focus in the shape of blobs. In figure 5.3 we have applied the k–means method, and a small portion of the central part of the image is shown for better detail. (a) is the detailed input image. (b), (c) and (d) are the three clusters generated using this technique to the input image as it is. The last row are the respective clusters when the input data to the clustering algorithm is the spread sharpened image. Small differences can be appreciated but they can be crucial in describing textures based on blobs

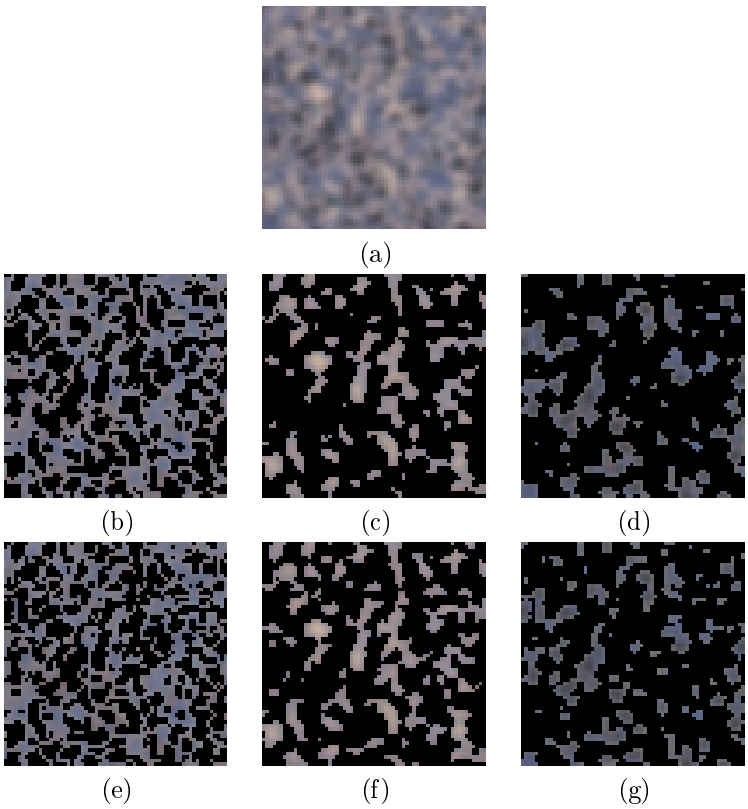


Figure 5.3: K–means clustering example: (a) is the central area from the original image to which clustering is performed. The following row is the result of segmenting in 3 clusters on the original image. Last row when applied to the sharpened image.

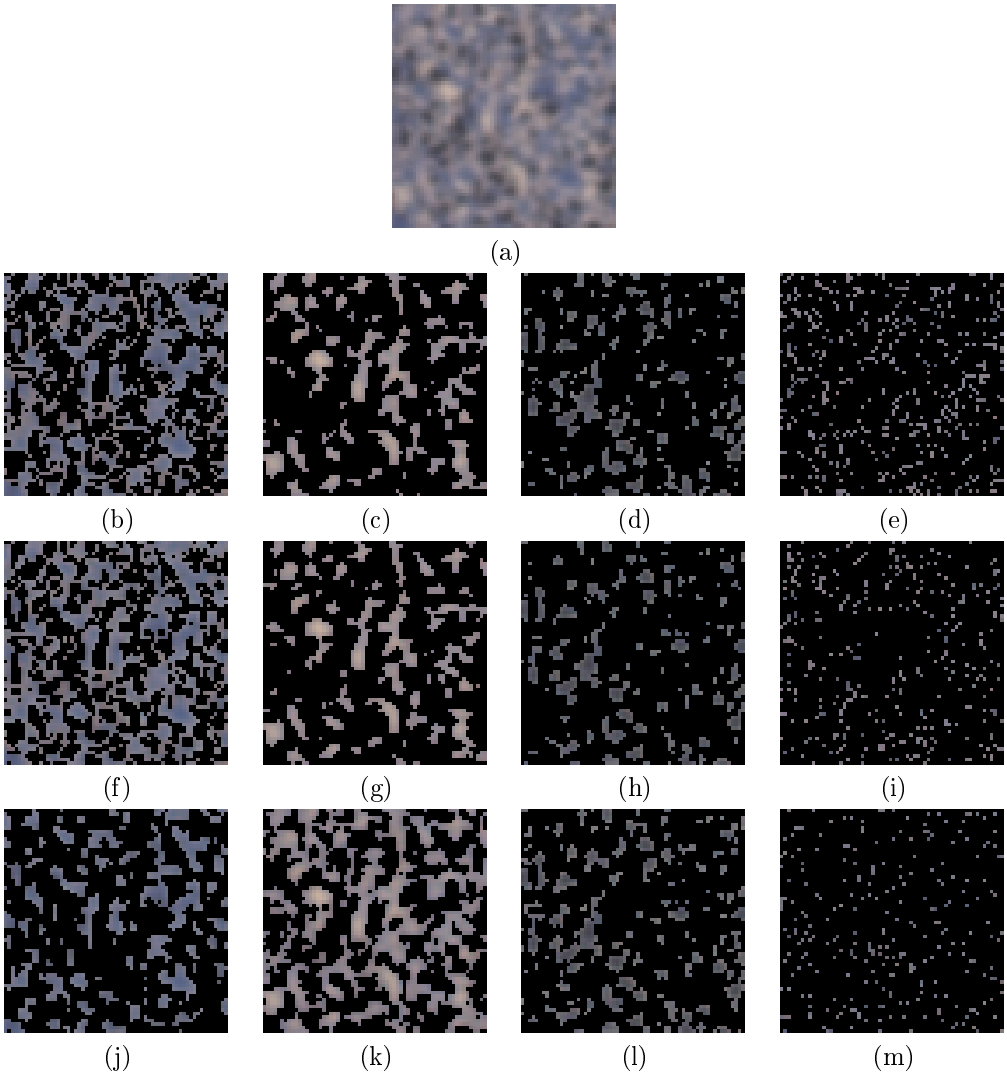


Figure 5.4: Mixture of gaussians clustering example: (a) is the central area from the original image to which clustering is performed. The following row is the result of segmenting in 3 clusters on the original image. The middle row when applied to the sharpened image. The last one when using automatic supervised clustering.

characteristics. In the middle and right cluster, the less populated, some blobs that are merged in the first row are well segmented in the last one. One of the advantages of this method is its simplicity and low computational complexity, however the result are not very good as colours rarely group in spheres. It is more likely to find colour distributions resembling gaussian distributions, and so mixture of gaussian fit well when it is the case. The following picture is from the same image but applying the expectation maximisation mixture of gaussian distributions. Figure 5.4 are the resultant images whit this clustering method. The first row of images contains the image of clusters from the original image and the last image of the row with those points that are classified as *ambiguous cluster* following equation 5.7. It has already improved the clustering without using the perceptual sharpening. The second row of images has the same configuration being the results of the sharpened image. Some characteristics are remarkable, the number of ambiguous pixels reduces in this case and like in the k–means algorithm blobs are better segmented in the case of dark and light blobs. Blue blobs are somewhat messy and is difficult to appreciate improvement. Even in this case when analysed carefully the blob boundaries are better. The last row introduces a new approach to the clustering. As it has been commented these techniques can be supervised initialising the first guess on the centres of the clusters, normally it is done manually. This supervision can be done automatically if some maximum localisation criterion is used. However, in our case the colour data forms a 4–dimensional distribution surface. To construct such surface is very time and space consuming and it is unfeasible. The approach used is to locate the desired number of maximums in a 3D space projecting one of the dimensions and use it to find compatible maximums in the other projections. These maxima are the input centres of the mixture of gaussians. To control the behaviour of the EM algorithm we only iterate on the covariances, fixing the mean of the gaussian distributions. Applying this criterion we obtain the results of the third row of figure 5.4. The density of blobs is more equilibrated and principally, the blue cluster contains more homogeneous colours, and the blobs are more accurate. The light blobs are the ones which receive the colours that are not included in the blue cluster. Nevertheless the inhomogeneity is not very large, in fact when looking at the original image white points are more dispersed. It should be noted that this criterion can not be done in the original image because there not exist clear maxima, whereas in the sharpened image it does.

5.3 Global features

With the following global colour measurements we try to capture a first coarse description of the image using basic statistics. For a given image $\mathbf{I}$ of size $N \times 3$ where each row is a colour triplet and p different types of blobs we will derive from the clustering step the images $\{\mathbf{I}_1, \ldots, \mathbf{I}_p\}$.

Global colour mean: A global colour measurement of the whole image:

$$M_1(\mathbf{I}) = \bar{\boldsymbol{I}} = \frac{1}{N}(\boldsymbol{I}'\mathbf{1}) \,, \tag{5.8}$$

where $\mathbf{1}$ is the constant 1 vector of dimension N.

Global colour variance: A global colour measurement of how constant the colour on the whole image is:

$$M_2(\mathbf{I}) = \frac{1}{N-1} diag((\boldsymbol{I} - \bar{\boldsymbol{I}})'(\boldsymbol{I} - \bar{\boldsymbol{I}})) \tag{5.9}$$

where $diag(A)$ is the diagonal vector of the matrix A.

Obviously these two measures describe very coarsely the colour contents of all the surface.

5.4 Local features

Now we will give some measurements on local properties of the blobs. We are defining the measures to get a more detailed description of the geometry and distribution of colour textons. Most of them are based on the central moments of inertia ([60]), which are defined as:

$$m_{p,q} = \sum_{(x,y)\in\mathcal{R}} (x - \bar{x})^p (y - \bar{y})^q \tag{5.10}$$

where $\mathcal{R}$ is the region (or blob) of the image where the moment is calculated, and $\bar{x} = 1/n \sum_x x$ and $\bar{y} = 1/n \sum_y y$ is the centre of gravity with n the total number of pixels in the region. The moments are calculated for each blob in each segmented image. Next, we enumerate the different moments proposed to the description of the form of textons. Although there are many more possible, using high order moments will capture information on blob geometry, which is usually useless when dealing with a large amount of blobs. All this measures are over a binary mask of the colour segmented images.

Blob area It is the moment of order (0,0) that simplifying is the count of the pixels of a region.

$$a(\mathcal{R}) = m_{0,0} = \sum_{(x,y)\in\mathcal{R}} 1 \tag{5.11}$$

Blob eccentricity A measure of how rounded a region is. When the blob is line–shaped the value of this measure, ε, is 1 and 0 when it is circular.

$$\varepsilon(\mathcal{R}) = \frac{(m_{2,0} - m_{0,2})^2 + 4m_{1,1}^2}{(m_{2,0} + m_{0,2})^2} \tag{5.12}$$

Elongation A measure proportional to the elongation of the object taken in the direction which maximises the measure. It can be set as the maximum eigenvalue.

$$\lambda_{max}(\mathcal{R}) = 2\sqrt{\frac{m_{2,0} + m_{0,2} + \sqrt{(m_{2,0} + m_{0,2})^2 + 4m_{1,1}^2}}{2m_{0,0}}} \tag{5.13}$$

Orientation The angle, ϕ, between the x axis and the axis around which the blob can be rotated with minimum inertia which is given by the eigenvector to the minimal eigenvalue.

$$\phi(\mathcal{R}) = \arctan \frac{2m_{1,1}}{m_{2,0} - m_{0,2}} \tag{5.14}$$

Instead of recording all the values for each blob in each image the information is reduced to the mean and standard deviation of the measures. For every segmented image $\mathbf{I}_i$ we obtain a set of parameters to define the overall form of blobs in it:

$$\begin{aligned} M_1^i &= \frac{1}{N^i} \sum_{b \in Reg(\mathbf{I}_i)} a(b), \\ M_2^i &= \frac{1}{N^i} \sum_{b \in Reg(\mathbf{I}_i)} \varepsilon(b), \\ M_3^i &= \frac{1}{N^i} \sum_{b \in Reg(\mathbf{I}_i)} \lambda_{max}(b), \\ M_4^i &= \frac{1}{N^i} \sum_{b \in Reg(\mathbf{I}_i)} \phi(b) \end{aligned} \tag{5.15}$$

which are the mean values, being $Reg(\mathbf{I}_i)$ the set of blobs in the segmented image $\mathbf{I_i}$, and N^i the number of blobs in the image. And the measures standard deviations:

$$\begin{aligned} M_5^i &= \frac{1}{N^i - 1} \sqrt{\sum_{b \in Reg(\mathbf{I}_i)} (a(b) - M_1^i)^2}, \\ M_6^i &= \frac{1}{N^i - 1} \sqrt{\sum_{b \in Reg(\mathbf{I}_i)} (\varepsilon(b) - M_2^i)^2}, \\ M_7^i &= \frac{1}{N^i - 1} \sqrt{\sum_{b \in Reg(\mathbf{I}_i)} (\lambda_{max}(b) - M_3^i)^2}, \\ M_8^i &= \frac{1}{N^i - 1} \sqrt{\sum_{b \in Reg(\mathbf{I}_i)} (\phi(b) - M_4^i)^2} \end{aligned} \tag{5.16}$$

In some cases it can be useful to get a more fine description of the distribution of a certain measure. Then, instead of using the mean and the standard deviation we will use an small dimensional histogram of the parameter in question. For example, we have used in one of the cases that we will present in section 5.6, the histogram of blob areas for each segmented image. As we knew a priori which were the usual size of the blobs we divide them in four bins: very small, small, medium and large blobs. When the texture is thought to be described by a certain parameter it is straightforward to obtain its histogram. The small histogram will be denoted as the feature M_9^i where the number of bins is specific for each problem.

As we want to describe colour textures, a representation of the colour of each cluster is needed. Three more parameters are extracted from each image $\mathbf{I_i}$. We compute the mean colour of all the pixels in each cluster as $M_1(\mathbf{I_i})$

Although strictly speaking only M_1, M_2 and $M_1(\mathbf{I_i})$ are measurements on colour, all the other measures can be considered to extract information of similar colour blobs, defining, separately, how the texture of colours are. The colour space where the colour measurements are done is not specified. Any suitable one can be selected whereas it captures the information that is wanted. As in high frequency patterns colour assimilation is one of the main factors to perceive colour, a perceptual blurring as defined in [12] and introduced in section 4.2 is feasible. With this approach the distance from the scene to the observer can be modeled by a few parameters.

5.5 Proposal for a perceptual colour texture representation

So far we have introduced the methods to isolate the blobs on the image that form the texture based on the colour information. Colour induction has been used to enlarge colour differences and help to a better segmentation of coloured blobs. It is done at different scales to simulate the human process of attentive vision. When done at high frequencies perceptual blurring is performed, and at low frequencies we apply perceptual sharpening.

The second stage has been to define a set of measures to capture global an local features on the colour and texture of the image. At this point there is no connection between both stages.

Our proposal is to merge the two previous points in a single feature vector representing the colour texture descriptor of the image. The schema of this proposal is presented in figure 5.5. We will take the global measures from the set of images obtained from the assimilation process, and the local measures from the blob segmentations using different scales on the contrast process. The number of segments used in the local features will depend on the image content and, when available, on the a priori information of the problem.

From the schema of colour assimilation and contrast a set of images are obtained that have to be recombined to be the input to the feature extraction. Given a scale s_p, where $1 \leq p \leq n$,(i.e.: assimilation conditions) the outputs $A(Opp(I)_c, s_p)$ for $c = 1 \ldots 3$ are combined in a tristimulus image $A(I, s_p)$, that is, the image obtained after an assimilation process at observer conditions defined by s_p. For all the set of images obtained at the different scales for assimilation we compute the global feature vector composed by M_1 and M_2:

$$\{M_f(A(I, s_p))\} \;\; p = 1 \ldots n, \;\; f = 1 \ldots 2 \tag{5.17}$$

A similar process is done with the outputs from the perceptual sharpening, the images $C(Opp(I)_c, s_p)$ for $c = 1 \ldots 3$ and $n+1 \leq p \leq m$, are combined in a tristimulus image $C(I, s_p)$ and then a set of k segmented images is obtained from the clustering process, $\{C^{sg}(I, s_p)\}$ for $sg = 1 \ldots k$. At this point we have $k \times (m - n)$ images as the result of the clustering process on the $m - n$ scales of the perceptual sharpening.

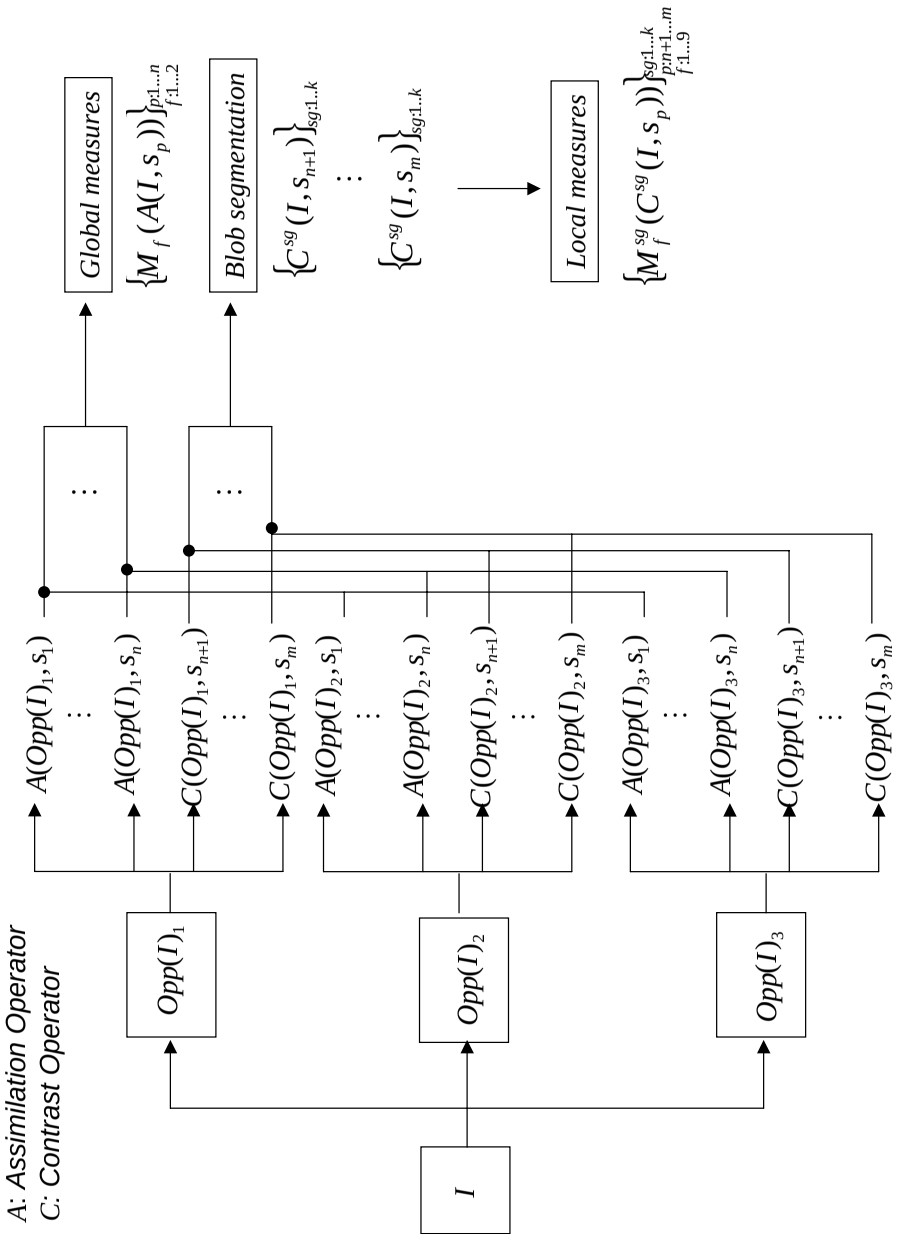
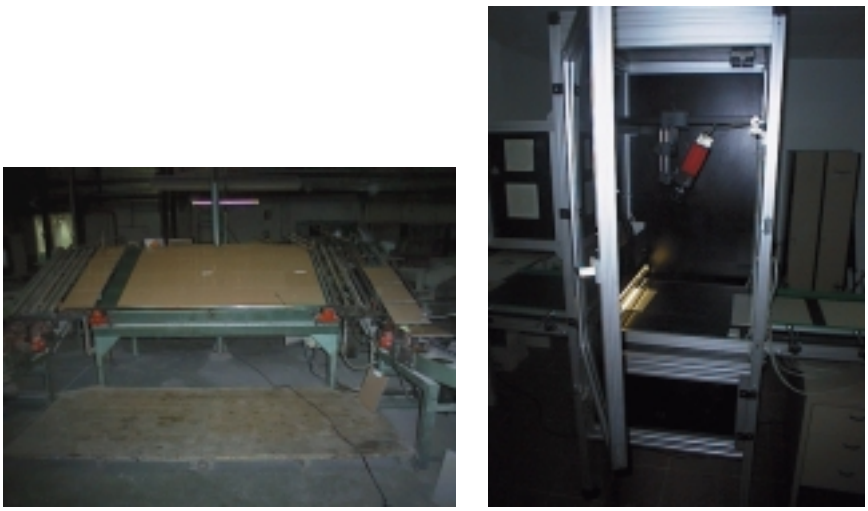


Figure 5.5: Colour–texture representation model.



(a) (b)

Figure 5.6: Tile classification system: (a) Real conditions in which human operators classify the production. In (b) the off-line classification system designed for this purpose.

From those images the local feature vector is obtained. The features included are those defined in section 5.4, from M_1^i to M_9^i:

$$\{M_f^{sg}(C^{sg}(I, s_p))\} \text{ where } sg = 1 \dots k,\ p = 1 \dots n,\ f = 1 \dots 9 \tag{5.18}$$

This is a general framework for colour–texture representation, but in each problem the parameters k, m, n, and the desired features f have to be defined to adjust to the knowledge of the problem.

From now on we will describe two applied cases were colour texture is the key factor to analyse the problem. The first case is the classification of ceramic tiles were a representation of the texture is needed to differentiate between classes with small differences in colour and/or texture. The second case is the quantification of printing quality from the appearance of the texture in homogeneous ink patches.

5.6 Case 1: Ceramic tile classification

5.6.1 The problem

In this section we treat a specific problem of classification of polished ceramic tiles. Tile manufacturing needs of pigments and clay which are mixed, melted, sprayed to form the tile substrate, and finally baked. This is a high quality tile whose production can be affected by external factors that are difficult to control, such as humidity, temperature, pressure conditions, origin of clays and colour pigments. Changes in

any of these parameters provoke subtle visual variations of the tile aspect when tiles are placed on the floor, one next to the other. These visual changes are due to small alterations of colour and texture properties of the tiles. It forces an on line classification of the production. At present, the classification is done by human experts, and it always involves subjectivity and loss of repetitiveness. In figure 5.6(a) we can see the place devoted to classify the production. In each production line only one model of tiles is produced. Thus, the classification must be done among classes of each model and not among models. During one day production up to eight classes can be created. A correct and non–subjective ceramic tile classification would allow to avoid returns from customers and to optimise the storage of the production stock reducing stock fragmentation. Previous research in computer vision techniques has contributed with interesting works on this problem [12, 83]. Now we will work on it with our colour texture description approach.

Previous works on the same application only use colour information for the classification task. In one of them [83] the first and second order moments of the RGB histograms are computed as colour and texture measurements, respectively. The second work [12] is based on three-dimensional histograms over the RGB space. The classification process extracts a similarity measure based on the Pearson correlation coefficient between 3D histograms. None of them compute blob measurements.

5.6.2 Human criteria for tile classification

Nowadays, this classification task is performed by specialised workers requiring a training period before to do it. One worker is replaced from the production line every two hours in order to avoid fatigue. It is such a subjective task that two different people can disagree in classifying the same sample. However, they have developed their own jargon to speak about tile differences. With the collaboration of a company of this industrial sector we did an experiment with human operators om order to get maximum information on hoe they do the classification.

Firstly, we asked classification experts to list the vocabulary that has been the basis to develop this work. The following list presents the characteristics they look at:

1. Fine–grained vs. coarse–grained: It is an obvious feature that defines the size of the grains.

2. Opened grain vs. closed grain: it is a measure of the distance among grains of the same size that could be intuitively interpreted as some density factor.

3. Light vs. dark grain colour: The colour properties of a specific type of blobs.

4. Light vs. dark background: The colour properties of the tile background. For some tile models with an important difference between the amount of every colour can provoke a predominant colour and a secondary colour. The first one will be called the background and will the blobs froming this bakground will not be considered.

5. Light vs. dark global colour: The colour properties of the overall colour impression due to the interaction between background and grains or between all grains.

Two tests were done with a group of trained people to search for the motivations they make conclude a certain classification.

- For a set of seven different classes of the same model and for two models they were asked to enumerate and quantify which were the features that, from their point of view, allowed to discriminate between a pair of classes. When describing these differences several tiles of each class were used.
- For each class the expert had to determine which were the three most similar classes.

The conclusion in the first experiment was that they do not focus on the interaction between colours but in the global appearance of tile and in local features of each colour. In the second case it was clear that they admit that some classes can be mixed without a very high inter–class difference, and that when a class can be confused with another one the behaviour is usually symmetric. From the models used in the experiments and from the experience of human operators, it was agreed that human classification is more difficult as the number of colours increases.

5.6.3 Preliminary approach

Although there exist some commercial systems that claim they cope with the ceramic classification problem only the works in [83] and [12] are documented, the others only have a short brochure without any indication of its classification rates and the methods they use. In both cases global colour information has been used, although the influence of the texture has been considered by simulating an assimilation step in the second one.

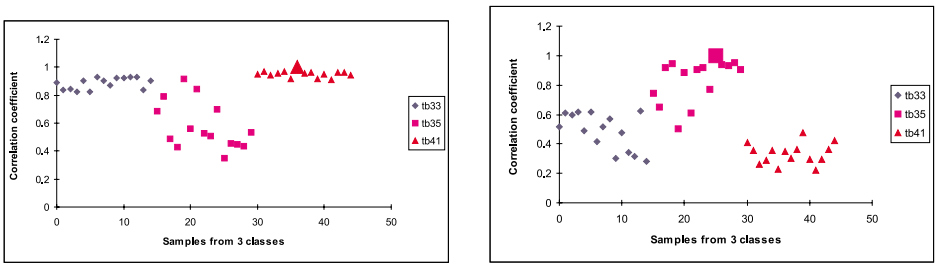
Our first attempt is to check if using only global colour information is enough or something else is needed.

Similarity measurement

In computer vision the problem of defining similarity measurements has been widely studied for the object recognition task. In [27] it is argued that the recognition is mainly based on distances on a small dimensional space and on the definition of prototypes which can define a class of objects.

In this work, as in [13] we will use the linear correlation coefficient (*Pearson's r*) between image histograms to compare the tiles. Then we define D as a similarity measurement:

$$D(r,s) = 1 - \left| \frac{\sum_i (H(I^r)_i - \overline{H(I^r)})(H(I^s)_i - \overline{H(I^s)})}{\sqrt{\sum_i (H(I^r)_i - \overline{H(I^r)})^2} \sqrt{\sum_i ((H(I^s)_i - \overline{H(I^s)})^2} \right| \tag{5.19}$$



(a) (b)

Figure 5.7: Classification of the same tile set using two different reference tiles: (a) Reference tile is 36. There is no possibility to discriminate between classes tb33 and tb41. (b) Reference tile is 26. Only class tb35 can be discriminated in relation to the other two.

where I^r and I^s are images to be compared, $H(I^j)$ is the histogram of the I^j image and $H(I^j)_i$ denotes de number of pixels of the I^j image having colour i, where i is a triplet of red, green and blue values which range from 0 to 2^b, being b the number of bits used to digitise the image. $\overline{H(I^r)}$ is the histogram mean.

So that, the correlation coefficient helps us to get a similarity value between two different tiles. Every tile is defined by its own histogram. If $D(r,s)$ value is near to 0 it means that images I^r and I^s belong to the same class, on the contrary, a value near to 1 signifies images are from two different classes.

Classification by similarity

The use of similarity measures to classify involves the election of the representative samples for every class of tile. Usually, they are arbitrarily selected among those whose classification is known. This can cause some problems since an algorithm can obtain different results depending on which is the sample used as prototype.

In Boukouvalas's work [13], the classification uses one reference sample. The algorithm classifies based on thresholds on the correlation coefficients between the sample and the reference tiles that have been taken. Applying this method to our samples makes us realise on how important is the choice of the reference sample. Changing the reference sample can vary the capability to discriminate among classes. An example of this problem is shown in figure 5.7. The x axis is the number of sample and the y axis is the distance $D(r,s)$ from each sample to a reference sample, which is different in the first and second plot.

To avoid the above mentioned problem and to get a classification less dependent on the reference chosen, we propose a new method to select the prototypes of each class. N images of each class of tiles were taken to elect the representative sample. From them we obtained a similarity matrix calculated using the measure D between the three–dimensional histograms of all of them.

To select the most representative images for every class, its configuration inside the space of classes has to be analysed. We used *Multidimensional Scaling* as a

method to explore the configuration among them [64]. This method allow us to use a similarity matrix to find the point coordinates of each sample in a $\mathcal{D}$–dimensional space. The distances between pairs of points in the configuration agree with the similarity measurements.

The dimension $\mathcal{D}$ of the space where samples are represented is fixed by the stress measure. This is an error measure between the similarity matrix used and the distances matrix of the points in the new space. In our experiment we generate a 6–dimensional space with an stress value of 0.07, which we consider enough for our purposes. A stress value of 0.05 is considered good to establish the real space dimension.

In this space, we group the samples of the same class. For every group, the first three samples closest to the centroid of the group are taken as class prototypes.

The classification process consists of two steps. Firstly, to calculate the similarity of the input sample to every prototype computed in the above step. Secondly, if the minimum distance does not exceed a certain threshold, $\mathcal{T}$, the input sample is classified as belonging to the same class of the representative sample from which the minimum is obtained. Otherwise a new class is defined and a new space to classify is calculated. The parameter $\mathcal{T}$ determines the stock fragmentation, the smaller $\mathcal{T}$ the higher fragmentation.

Results

The test has been carried out in laboratory conditions, where 90 tiles were used from 3 different models with 3 classes in each model. We used the full histogram to calculate the similarity D implementing the histogram with BTrees in order to save memory space. If the histogram dimension is reduced the method is unfeasible because of the small differences in colour between classes. The classification rate was over 90% using exclusively colour information, but when used in a larger set of samples the classification rates dropped to approximately 75%. This was because colour can not cope with all the cases and both colour and texture has to be considered. There are samples with the same colour distribution that they only differ on how colours are distributed on the sample. These were the first steps in the colour texture inspection problems.

5.6.4 Classification based on proposed perceptual features

From the conclusion of section 5.6.2 we translated the list of visual features to computational features on the image. There is not a one to one equivalence between expert and computer features but we define measures that involve several expert words.

We can associate measurements M_1^i, M_5^i and the small dimensional histogram on $a(\mathcal{R})$, M_9^i, to the first characteristic from the list. M_9^i can also catch information on the second characteristic. Light vs. dark grain colour and Light vs. dark background are represented by measurement $M_1(\boldsymbol{I}^i)$, depending on the segment applied it will be grain or background. Which one it is the segment assigned to background is not relevant because we do not need to explain the reasons because the computational system classify, and for all i, $M_1(\boldsymbol{I}^i)$ will be calculated. $M_1(\mathbf{I})$ and $M_1(\mathbf{I})$ are used

to define the fifth characteristic, Light vs. dark global colour. We also add two more measure to the list of selected ones, M_2^i and M_7^i because when the *grain is open*, in their jargon, blobs are more rounded and have less variation in its elongation. From this set of measures M_1^i and M_5^i are rejected because they are redundant to some extent with M_9^i.

In short and as a summary 2 vectors are obtained as a more global description ($M_1(\mathbf{I})$ and $M_2(\mathbf{I})$). Two more set of vectors describe globally how are the main statistical properties of each of the colours in the image, $M_1(\boldsymbol{I}^i)$ and are applied to each of the p segmented images. Finally the M_9^i, M_2^i and M_7^i measures geometric properties of the blobs in each of the images $\boldsymbol{I}_i$.

To result these concepts in the general colour-texture representation (section 5.5), parameter f associated to global features is $f = \{1, 2\}$, and f associated to local features is $f = \{2, 7, 9\}$ The last step is to input the feature vector into a classifier. We have used the Linear Discriminant Analysis defined in A.

Results

We have tested the classification method on a variety of feature vectors obtained combining the two clustering methods and applying perceptual blurring and without it, that is: $n = 0$ and $n = 1$ in the colour texture representation model. The segmentation method is done on the original image and in the perceptually sharpened image, $m = 0$ and $m = 2$. We did not combine them, but tested two different configurations $\{n = 0, m = 0\}$ and $\{n = 1, m = 2\}$. As the tiles of the same model have a single main frequency, there is no need to extent the feature vector to a multiscale representation on perceptual blurring and sharpening. For both cases, the frequencies are selected manually for each model.

We have used six different models of tiles with a total set of 514 samples distributed in 47 different classes. Each sample has been divided in three regions which results in 1542 images. One third of the images are randomly selected to be used as learning set and the others are the test set. The classification percentages obtained are those from table 5.1 to 5.6. The first column of the tables is the preprocessing applied to the image which can be the SS operator ($\{n = 1, m = 2\}$) or the image itself ($\{n = 0, m = 0\}$). The second column is the clustering algorithm used where MG stands for Mixture of Gaussians. As the experts recognise that some times the same sample can be included in two different classes, we have obtained the percentage of images classified as the class that they actually belong and the percentage of images that their real class is the first or second best match. Due to the fact that the tiles are previously sorted by humans it is reasonable to take this second percentage as the capability of the system to cope with the problem. The percentages are taken only on the test sample, the learning set is not considered.

Not all the models have the same number of pigments. When it increases the human experts have more difficulties to classify them. The models used in this experiments where Duero, Tiber, Cinca, Orinoco, Ohio and Mijares with 2, 4, 3, 3, 2, and 3 different pigments respectively. Table 5.7 contains the global results for all the models. The conclusions from the tables can be summarised in

- When the number of pigments is greater than two, there is more confusion

Process	Clustering	1st rank	1st or 2nd rank
SS operator	MG	98.75%	100%
	KMeans	97.08%	100%
None	MG	84.16%	92.08%
	KMeans	85.83%	92.91%

Table 5.1: Classification results for Duero model. Number of images in the test set: 240.

Process	Clustering	1st rank	1st or 2nd rank
SS operator	MG	97.5%	100%
	KMeans	97.08%	99.5%
None	MG	92.5%	96.66%
	KMeans	91.25%	96.66%

Table 5.2: Classification results for Tiber model. Number of images in the test set: 240.

Process	Clustering	1st rank	1st or 2nd rank
SS operator	MG	98%	100%
	KMeans	98%	99%
None	MG	89%	92%
	KMeans	87%	93%

Table 5.3: Classification results for Cinca model. Number of images in the test set: 100.

Process	Clustering	1st rank	1st or 2nd rank
SS operator	MG	94.05%	99.01%
	KMeans	91.08%	97.02%
None	MG	88.17%	90.59%
	KMeans	86.13%	89.10%

Table 5.4: Classification results for Orinoco model. Number of images in the test set: 202.

Process	Clustering	1st rank	1st or 2nd rank
SS operator	MG	99.30%	100%
	KMeans	99.30%	100%
None	MG	95.83%	97.22%
	KMeans	94.44%	96.52%

Table 5.5: Classification results for model Ohio. Number of images in the test set: 144.

Process	Clustering	1st rank	1st or 2nd rank
SS operator	MG	100%	100%
	KMeans	100%	100%
None	MG	100%	100%
	KMeans	100%	100%

Table 5.6: Classification results for Mijares model. Number of images in the test set: 102.

Process	Clustering	1st rank	1st or 2nd rank
SS operator	MG	97.47%	99.61%
	KMeans	96.40%	99.03%
None	MG	90.76%	94.36%
	KMeans	89.30%	93.87%

Table 5.7: Average classification results for all models.

between blobs and working on the original image has worst results.

- Although the ratios are not so high than in the previous case, perceptual sharpening operator always gives better results even if the number of different pigments is low.

- The use of the clustering algorithms is not very crucial in the final classification. There are some advantages in the case of mixture of Gaussians, but at the expense of processing time and memory usage. Thus, when the classification ratio is not crucial *kmeans* is a feasible option.

- Due to subjectiveness in the human classification, taking the first and second best classes to fit the sample is an approximation in the uncertainty introduced in the classification process.

Although the number of classes in each model is important, the higher the number of classes is the higher probability of misclassification, but it can happen even with very few classes. In the case of Cinca model there are three different classes but the classification is not perfect, whereas in the Mijares model there is no error with the same number of classes. This is because of the the tiles supplied in the second case are very different between them and any approach will solve it. In regard to Duero, Tiber, Orinoco and Ohio models, the number of grades respectively is 10, 12, 11 and 7.

In spite of these good results, the problem is not completely solved. What we get is an evidence that the colour texture description approach is valid for this problem. However the dynamic creation of new classes inside a model is not obvious, and until it is not done a full on–line system can not be build. To work with this problem we should have all the tiles produced before and after a new class is created, and that needs a more complex logistic when the prototype is done in off–line production.

None of the works found in this field cope with this problem and it is still an open problem to solve in the future.

5.7 Case 2: Printing Quality evaluation

The second industrial problem that we focused on was a particular aspect of the printing quality evaluation. One of the problems of the printing industry is to assure printers can print homogenous colour patches, if they cannot do it very poor final results are obtained. Sometimes the print head produces a vertical degradation that translates to an striped patch which is called a banding effect that diminishes the printing quality. Sometimes it is a very smooth effect and sometimes is very obvious. As many other applications it is done by humans, which implies subjectivity and non–repetitiveness, as in the previous case. To control the printing quality they grade each patch in different levels. The problem is to find a set of parameters to automatically evaluate this grade or, even better, to give a number in a continuous domain of how good the printer is.

Unfortunately we had a very short set of examples. In addition the images were scanned at a medium resolution and we do not know exactly the parameters of the process. Despite these adverse circumstances we have tried the SS operator to this problem.

The perceptual sharpening operator SS can work on any colour space whenever it is tristimulus. Although the opponent space is the best to be used when simulating human behaviour, others can be used to make differences more obvious in specific problems. That is relevant when analysing printed images because the tristimulus used in this process is based on the subtractive primaries *cyan*, *magenta* and *yellow* plus a fourth ink introduced for practical purposes, *black*, and so it is called CMYK model. Then instead of using the opponent space we will use the SS operator on the CMY space as the quality test are usually done with pure inks. With this modification we can operate only on the principal channel of the patch, which is a priori known. For example, if we operate on the opponent space a nearly homogeneous cyan patch will present a colour distribution that will be shared by the two chromatic opponent channels. When performing in the CMY space the representation lays on a single channel and this is the only one that has to be sharpened.

The transform from RGB (the original colour space of the images) and the CMY space can be approximated by:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{5.20}$$

When testing the black ink, as it is just luminance, the opponent space is used and the first component is sharpened as there should be no chromatic information.

The images from banding effects are somewhat textured because the white colour of paper sheet becomes more visible as worst is the defect of the printer. That is, the more evident is the defect the more texture appears in the image. A printed image is never perfectly homogeneous but when it is good the colours that appear are very

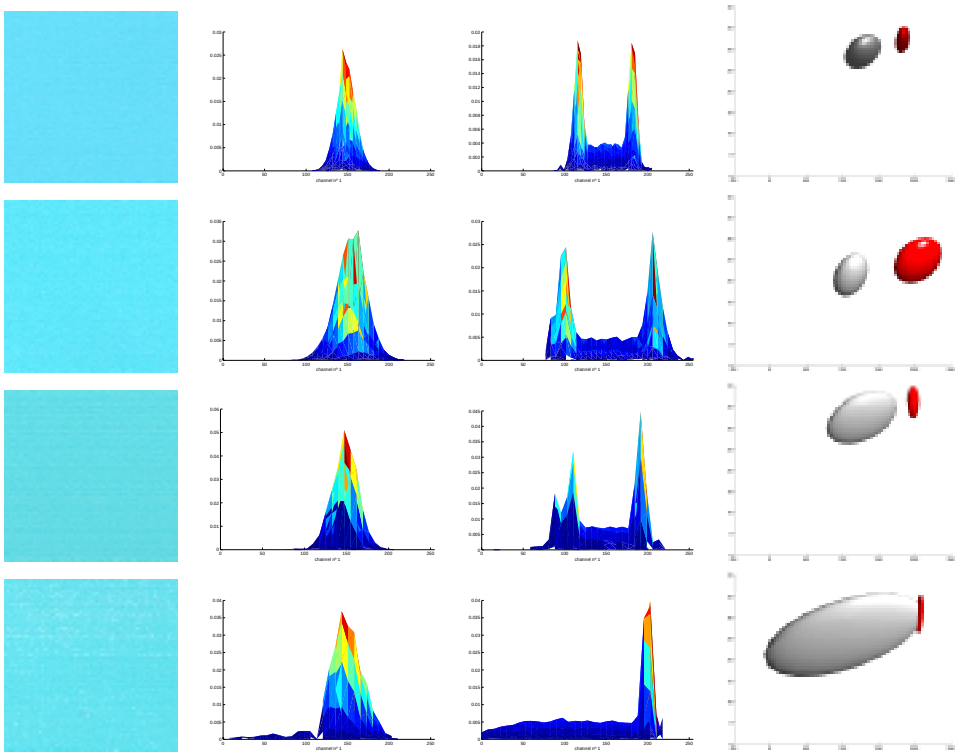


Figure 5.8: Cyan patches from different printers, ordered from top to down by its banding quality. From left to right: original image, colour distribution of the image on the cyan channel, distribution of the image after the SS operator in the CMY colour space, gaussian mixture of the previous distribution.

close one to the other. When there exist non–homogeneity and the colour distribution is analysed the main colours that appear are white (the paper colour) and the hue of the basic ink. Those colours are clearly far away one from the other. We will use this effect to visualise the printer banding error.

What we do is to cluster the colours of the image into two groups using the decision criterion based on modeling de distributions by a mixture of gaussians defined in section 5.2.2. If the region is homogeneous two clusters with their means very close have to be found, and the covariance matrices of both of them have to be compact. One way to measure this compactness is to calculate the ellipsoids that includes 99.5% of the distribution. The length of the semi–axes is a parameter of how scattered is the colour. If a simpler parameterisation is needed the linear discriminant analysis of gaussian distributions can be calculated and then the mean and covariance will be 2–dimensional instead of 3–dimensional.

It is common that on some inks the banding effect is stronger than on others and this makes each ink to have different grades. For example, gray patches are

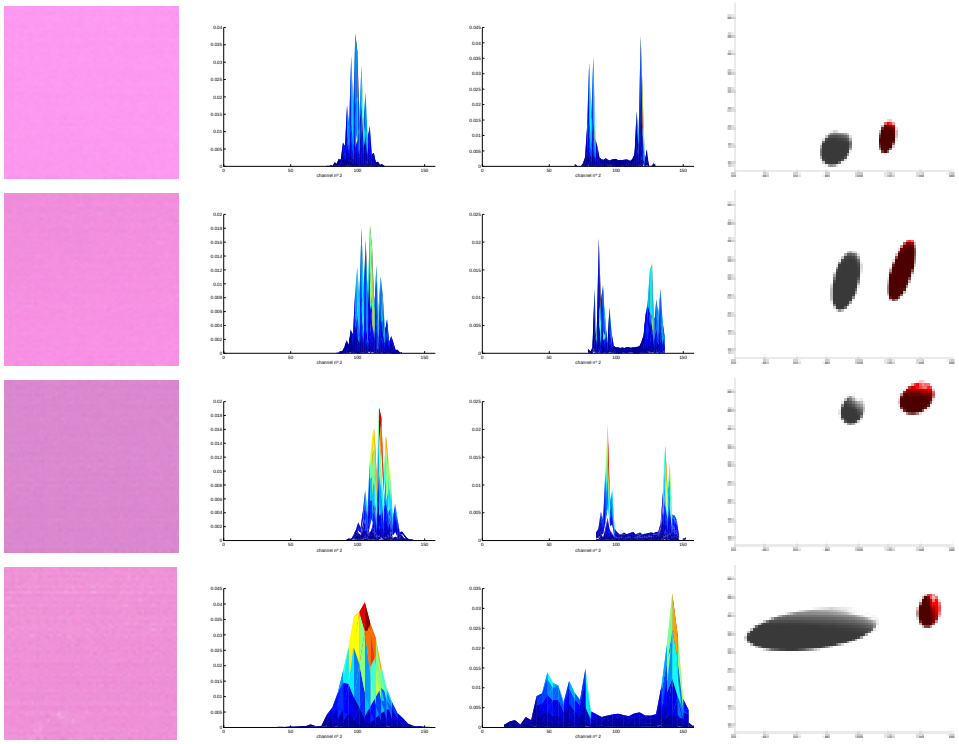


Figure 5.9: Magenta patches from different printers, ordered from top to down by its banding quality. From left to right: original image, colour distribution of the image on the cyan channel, distribution of the image after the SS operator in the CMY colour space, gaussian mixture of the previous distribution.

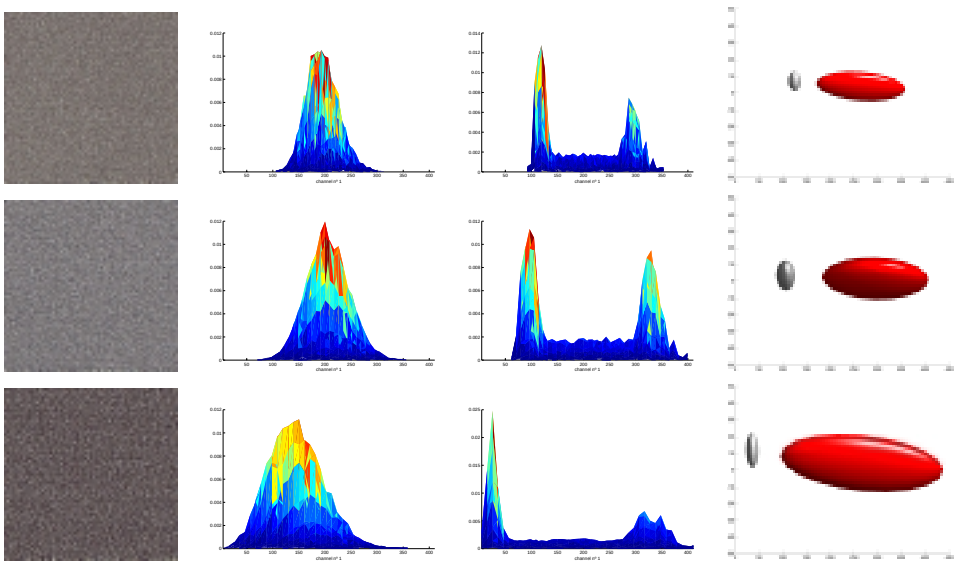


Figure 5.10: Grey patches from different printers, ordered from top to down by its banding quality. From left to right: original image, colour distribution of the image on the intensity channel, distribution of the image after the SS operator in the opponent colour space, gaussian mixture of the previous distribution.

usually better than cyan, and light colours are less sensitive to banding effects than the darker ones of the same hue. It is for that reason that in the examples that we show de number of grades are different and the magnitude of the parameters is specific for each ink.

Figures 5.8, 5.9 and 5.10 shows the results on some samples. The first one, a cyan patch, has four grades shown in the first column of the figure, sorted from best to worst according to the given classification. The first patch is clearly homogeneous whereas the last one is the worst case. In the middle the degradation can be continuous. When plotting the histogram of the cyan channel from the original image, second column, we observe a peak in all cases and in the last one there is a queue that is the white pixels. The other three cases are difficult to evaluate from this distribution. When applying the SS operator in the same view, third column, there appear two peaks when the image is more or less homogeneous, corresponding to the cyan in the image and the last one with one peak of the cyan and a more exaggerated queue of the white. The distance between peaks, if they exist, and their width can characterise the homogeneity of the sample. The results from modeling them with a mixture of gaussians are shown in the fourth column. The first ellipsoid, in red, is linked with the highest cyan in the image and the grey one to or the second cyan in the image or the white pixels. When it captures information of white pixels it is more elongated and its centre is far away from the centre of the red ellipsoid. If it is the coverage of a second cyan then the semi–axes are more similar and the centres are closer.

Figure 5.9 shows the same effect in a magenta sample where the same rules can be applied. And the last image is from a grey patch, where the red ellipsoid holds information of white pixels and the grey one includes the black pixels. In this case there are only three different samples because this colour is not so sensitive to the effect.

No numerical results can be extracted as we would need a number of samples to get a good validation of the parameters and their meaning. Despite this lack the preliminary results show encouraging performance towards its quantification.

5.8 Discussion

In this chapter we have addressed the colour texture representation based on perceptual mechanisms. To define a general framework to describe an image we have taken a multiscale approximation. In this way human attentive processes can be represented computationally. A clustering process on the perceptual sharpened images is done to better adjust the blob segmentation, which can be adjusted to reject ambiguous colours.

Another point of interest is the definition of two sets of features valid for a wide set of different sort of iamges. Global and local features are calculated for every scale needed. The use of feature vectors is not new in representing textures. What it changes from existing works is the use of the features on the images that segregate similar colours.

The general framework is closed defining a general schema that can be adapted to the knowledge of the problem, rejecting features, assimilation scales or contrast scales. Its flexibility allows adapting to a wide variety of problems.

Two of such problems are presented at the end of the chapter. The first one to demonstrate the viability of a classification system of ceramic tiles, which is badly posed as a problem of just colour differences. The second one uses the perceptual segmentation to quantify the banding error of printing industry. In this case, the results are preliminary but they present a good perspective.