Object recognition techniques in real applications
Fernandez Robles, Laura
Chapter 5

Object recognition for content-based image retrieval

5.1. Evaluation of clustering configurations of SIFT features for object recognition applied to CBIR

In this section we evaluate different techniques to determine when there is a correspondence between images and to compute the strength of the correspondence. On the one hand we use the similarity of the closest pair of keypoint descriptors. On the other hand we use a Hough transform to identify clusters of at least three points voting for the same pose of an object and we verify the consistency of the pose parameters with the least squares algorithm. We use different values for the Hough transform parametrization.

5.1.1. Method

We obtain SIFT keypoints and descriptors for the query object images and for all images of the dataset. Then for each query image, we compute the cosine similarity between a descriptor of the ROI with all descriptors of the query image. For this ROI descriptor, we consider the match that obtains the maximum similarity (minimum cosine angle) as long as its cosine angle is less than 2 times the cosine angle of the second nearest neighbour. Otherwise, we discard that match. Repeating this computation for all descriptors of the ROI, we obtain a set of matches between a ROI and a query image. Afterwards, we either use directly this information or we perform a voting and a geometric verification for pose of the object to decide about the correspondence between images.

On the one hand, we consider the correspondence of the match that achieves the minimum cosine angle among all matches between ROI and query image after the second nearest neighbour test. The pair of keypoints with the smallest angle is the most similar one among all pairs of matched keypoints and therefore this match has the highest probability of being correct. We use the value of such cosine angle of the most similar pair of keypoints as a measure of the similarity between the ROI and the query image. The hit list is ranked by sorting the retrieved images in ascending
order in relation to this metric. We refer to this case as without clustering.

On the other hand, from the set of matches between the ROI and the query image we identify clusters of keypoints that vote for the same pose of an object using Hough transform and we perform a geometric verification using least squares algorithm as suggested by Lowe (2004).

Each SIFT keypoint specifies 4 parameters: 2D location, scale and orientation. We keep track of these parameters for the match keypoints. Therefore, we can create a Hough transform entry predicting the model location, orientation, and scale from the match keypoints. The Hough transform creates a four dimension accumulator and uses each keypoint set of parameters to vote for all object poses that are consistent with it. When clusters of keypoints vote for the same pose of an object, it is more probable that they belong to the same object than just relying on a single keypoint (Lowe, 2004). Each keypoint match votes for the 2 closest bins in each dimension to solve the problem of boundary effects in bin assignment. Lowe’s clustering uses broad bin sizes of 30 degrees for orientation, a factor of 2 for scale, and 0.25 times the maximum projected training image dimension (using the predicted scale) for location. We refer to this case as Lowe’s clustering.

Afterwards, we use least squares algorithm to seek for geometric verification. We require each match in a cluster to agree within the Hough model, otherwise we consider that match as an outlier and it is removed. If less than three keypoints remain after discarding outliers, we reject the whole cluster of matches.

Finally for each remaining cluster, we compute the average of the cosine angles of the matches within the cluster. We take the minimum average for all clusters as a measure of the similarity between the ROI and the query image. Again, the hit list of retrieved images is sorted in ascending order according to this metric.

We evaluate other choices of the parameters used in the Hough transform model. We aim at obtaining a less restrictive clustering of matches by broadening their size (so lowering the number of bins). By considering broader bins, more keypoints agree for the same object pose. At the same time, less false correspondences are rejected. Half and quarter clustering settings use 60 and 90 degrees for orientation, factor of 4 and 6 for scale, and 0.5 and 0.75 times the maximum projected training image dimension for location respectively.

5.1.2. Evaluation

Dataset

For the purpose of ASASEC, retrieving objects from a dataset containing child pornography, we have created and made public our own dataset. It is composed of 614 frames of 640×480 pixels that come from 3 videos. All videos were recorded in

\footnote{dataset is available at http://pitia.unileon.es/varp/galleries}
5.1. Evaluation of clustering configurations of SIFT features for object recognition applied to CBIR

Figure 5.1: Examples of images containing the same object, a blue toy car. Changes in pose, scale, orientation, illumination and cluttered background can be noticed making the object retrieval task very challenging.

different bedrooms with different distributions, illumination, textures, etc., making the object retrieval a challenging task, Fig. 5.1. Nevertheless some objects are present in all videos such as two toy cars, some clothespins, a stuffed bee, some pens, some cups or a child book together with a big doll. The doll is usually the principal actor in the videos and helps us to simulate partial occlusions of the objects and a more realistic scenario. Although these objects are present in every video, they do not appear in every frame. Together with them, other objects are unique in each bedroom. We also provide a ground truth indicating which objects are visible in each frame.

Experiments and results

As query objects we have used the book, the blue and yellow car, and the pink, blue and green clothespin shown in Fig. 5.2. The total number of query objects present among the 614 frames of the dataset and the size of the ROIs are specified in Table 5.1.

When dealing with object retrieval, it is important that the retrieved images are ranked according to their relevance to the query object instead of just being returned as a set. The most relevant hits must be in the top few images returned for a query. Recall and precision are measures for the entire hit list and do not account for the quality of ranking the hits in the hit list. Relevance ranking can be measured by
computing precision at different cut-off points, this is technically called precision at \( n \) or \( P@n \). Let \( h[i] \) be the \( \text{ith} \) hit in the hit list and let \( \text{rel} [i] \) be 1 if \( h[i] \) is relevant and 0 otherwise. For a hit to be relevant the query object has to be present in the image and correctly localised. Therefore, if the image contains the object but the correspondence is not within that object, \( \text{rel} [i] \) is 0. Then precision at hit \( n \) is:

\[
P@n = \sum_{k=1..n} \text{rel} [k] / n
\]  

Table 5.2 shows the results for the four clustering types: without clustering, quarter clustering, half clustering and Lowe’s clustering showing the precision at

Table 5.1: Description of the query objects. Number of images that contain each query object in the dataset of 614 images. Size of each object ROI in pixels.

<table>
<thead>
<tr>
<th>Object</th>
<th>Number of query objects</th>
<th>Size of the ROI (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
<td>115</td>
<td>305×334</td>
</tr>
<tr>
<td>Blue car</td>
<td>102</td>
<td>285×258</td>
</tr>
<tr>
<td>Yellow car</td>
<td>138</td>
<td>208×265</td>
</tr>
<tr>
<td>Pink clothespin</td>
<td>125</td>
<td>146×132</td>
</tr>
<tr>
<td>Blue clothespin</td>
<td>92</td>
<td>85×145</td>
</tr>
<tr>
<td>Green clothespin</td>
<td>42</td>
<td>68×59</td>
</tr>
</tbody>
</table>

Figure 5.2: ROIs of the query objects.
5.1. Evaluation of clustering configurations of SIFT features for object recognition applied to CBIR

Table 5.2: Precision at cuts of the query objects using different clustering parameters. Best results for each precision at \( n \) are marked in bold.

<table>
<thead>
<tr>
<th></th>
<th>Book</th>
<th>Blue clothespin</th>
<th></th>
<th>Blue car</th>
<th>Pink clothespin</th>
<th></th>
<th>Yellow car</th>
<th>Green clothespin</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P@40</td>
<td>P@50</td>
<td>P@60</td>
<td>P@70</td>
<td>P@80</td>
<td>P@5</td>
<td>P@10</td>
<td>P@20</td>
</tr>
<tr>
<td>Without</td>
<td>1</td>
<td>1</td>
<td>0.9</td>
<td>0.8</td>
<td>0.75</td>
<td>1</td>
<td>0.7</td>
<td>0.35</td>
</tr>
<tr>
<td>Quarter</td>
<td>0.85</td>
<td>0.82</td>
<td>0.77</td>
<td>0.7</td>
<td>0.66</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>Half</td>
<td>0.93</td>
<td>0.88</td>
<td>0.88</td>
<td>0.83</td>
<td>0.8</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>Lowe’s</td>
<td>1</td>
<td>0.96</td>
<td>0.85</td>
<td>0.83</td>
<td>0.79</td>
<td>0.8</td>
<td>0.4</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Without</td>
<td>1</td>
<td>1</td>
<td>0.75</td>
<td>0.57</td>
<td>0.43</td>
<td>0.8</td>
<td>0.4</td>
<td>0.25</td>
</tr>
<tr>
<td>Quarter</td>
<td>1</td>
<td>0.8</td>
<td>0.6</td>
<td>0.43</td>
<td>0.38</td>
<td>0.2</td>
<td>0.1</td>
<td>0.05</td>
</tr>
<tr>
<td>Half</td>
<td>0.8</td>
<td>0.9</td>
<td>0.85</td>
<td>0.7</td>
<td>0.55</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>Lowe’s</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.73</td>
<td>0.625</td>
<td>0.8</td>
<td>0.4</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Without</td>
<td>1</td>
<td>0.9</td>
<td>0.75</td>
<td>0.73</td>
<td>0.63</td>
<td>1</td>
<td>0.5</td>
<td>0.3</td>
</tr>
<tr>
<td>Quarter</td>
<td>0.8</td>
<td>0.7</td>
<td>0.55</td>
<td>0.47</td>
<td>0.38</td>
<td>0.2</td>
<td>0.1</td>
<td>0.05</td>
</tr>
<tr>
<td>Half</td>
<td>1</td>
<td>0.8</td>
<td>0.75</td>
<td>0.7</td>
<td>0.68</td>
<td>0.8</td>
<td>0.4</td>
<td>0.2</td>
</tr>
<tr>
<td>Lowe’s</td>
<td>1</td>
<td>1</td>
<td>0.85</td>
<td>0.7</td>
<td>0.65</td>
<td>1</td>
<td>0.7</td>
<td>0.35</td>
</tr>
</tbody>
</table>

...different hits.

Examples of the second, fifth and twentieth hit of the hit list for the blue car with the different clustering approaches are shown in Fig. 5.3.

The ROI of the book has well-defined corners that can produce distinctive keypoints easier to detect and match among the images of the dataset. Without clustering approach correctly retrieved the first 51 hits just relying on the strongest match. This is a good result considering that there are 115 images containing the book in all 614 images of the dataset. However for higher cuts in hit list, Lowe’s clustering and half clustering approaches obtained higher precision results.

As for the cars, although SIFT method computes the descriptors using gray level images, there are small differences in shape and patterns between the two cars. Regarding the blue car, without clustering yielded the best results for low cuts of the hit list and Lowe’s clustering did for high cuts. For the yellow car, without clustering, half clustering a Lowe’s clustering obtained similar results.

Clothespins introduce a more difficult task since less distinctive keypoints are present and their shapes are very alike among them and with some other clothespins of the dataset. Most of the keypoints were found in the metal wire, near the holes or in the outlines. Precision at 20 only reached 0.35, 0.25 and 0.35 for the blue, pink and green clothespins respectively. Without clustering configuration achieved better...
results for retrieving the blue clothespin, Lowe’s clustering performed better for the green one and both approaches obtained the same results for the pink one.

All in all Table 5.2 shows that without clustering approach is more convenient for high precision at small cuts and Lowe’s approach at high cuts of the hit list in this dataset.

In Fig. 5.4 we present examples of misclassified query objects that have a similar shape but different colours leading to mismatches. This is because SIFT is not invariant to colour. Only three mismatches among different cars appeared in all the experiments for the first 20 hits of the hitlist but up to 20 in the case of the clothespins. This could be solved using a colour version of SIFT (Van de Sande et al., 2010).
5.1. Evaluation of clustering configurations of SIFT features for object recognition applied to CBIR

Figure 5.4: Examples of mismatches due to very similar objects that mainly differ in their colours.

Background is also another source of mismatches. The pattern duvet of one of the settings leads to many non relevant but distinctive keypoints that locally described can look similar to other objects. For example, some patterns of the duvet are similar to the patterns of the yellow car. Moreover, background of the ROI can contain distinctive keypoints that produce correspondences with other objects or backgrounds. Figure 5.5 shows examples.

Figure 5.5: Mismatches produced by pattern duvet.
5.2. Adding colour description to COSFIRE filters

In this section we present colour COSFIRE filters. They are trainable keypoint detection operators which are selective for given local colour patterns that consist of combinations of colour contour segments. It is based on COSFIRE filters for grayscale images introduced by Azzopardi and Petkov (2013c). Moreover, colour COSFIRE filters also add invariance to background intensity.

5.2.1. Method with application for colour vertex localisation

Overview

Figure 5.6a shows an input image with 8 vertices. We consider the vertex enclosed in the yellow rectangle as a (prototype) pattern of interest. The rectangular region is known as a region of interest (ROI). This ROI is shown enlarged in Fig. 5.6b. The colour COSFIRE filter configured from this prototype will respond to the same and similar patterns regardless of the background. The prototype has been manually selected by a user.

In Fig. 5.6b, the three ellipses represent the dominant orientations in region of interest. The circle denotes that several dominant orientations are overlapped. We detect the lines by symmetric Gabor filters and describe the colour of these lines by averaging the pixel values of a support region around the centre point of the ellipses for each colour channel.

We compute responses both for line detection and colour description at the centres of the corresponding ellipses in an input image. The response of line detection is computed by applying Gabor filters. The preferred orientations and band-

![Figure 5.6: (a) Input image of size 180×161 pixels. The yellow square marks the ROI from which the colour COSFIRE filter will be obtained. (b) Enlargement of the ROI. The ellipses represent the support of line detectors that are relevant for the concerned prototype.](image)
5.2. Adding colour description to COSFIRE filters

widths of the Gabor filters and the locations at which we take their responses are automatically determined at the configuration of the colour COSFIRE filter by analysing the prototype pattern. Therefore, the filter only responds to the same (or very similar) local spatial arrangement of lines of specific orientations and widths as in the prototype pattern. We compute the response of colour description of each line segment as the average of the pixel values in a support area around the centres of the corresponding ellipses for each colour channel. We compute the colour response by a Gaussian kernel that measures the similarity between the colour descriptions of the prototype and the input image. Thus, the filter only responds to the same (or very similar) local spatial arrangements of colours as in the prototype pattern.

The response of a colour COSFIRE filter is computed by multiplying the responses of the line detection and the responses of the colour description achieved in the centres of the corresponding ellipses and combining all the multiplications. The response of a colour COSFIRE filter comes from a pixel-wise evaluation of a multivariate function. For that purpose, the responses of Gabor filters and the responses of Gaussian kernels at locations around a pixel are previously shifted to come to that point.

In the following sections we explain the automatic configuration of a colour COSFIRE filter and its application to an input image.

Configuration of a colour COSFIRE filter for vertex localisation

Detection of orientations. We build the colour COSFIRE filter from the responses of 2-Dimensional Gabor filters applied to each colour channel. Gabor filters allow lines or edges detection, depending on their configuration, discriminating frequencies and orientations. Filtering individually the three colour channels and then combining these three responses increases illumination invariance and discriminative power leading to a more accurate detection of the activations in the image than filtering a luminance channel (Van de Sande et al., 2010), as for example the gray level image.

We denote by \( g_{\lambda, \theta, \zeta, c}(x, y) \) the response of a Gabor filter of preferred wavelength \( \lambda \), orientation \( \theta \) and phase offset \( \zeta \) to a given colour channel \( c \) of the prototype image \( P \). Regarding the considered phase offset of the sinusoidal wave function \( \zeta \), the Gabor filter could be symmetric (\( \zeta \in \{0, \pi\} \)), antisymmetric (\( \zeta \in \{\pi/2, 3\pi/2\} \)) or an energy filter by taking a quadrature pair of symmetric and antisymmetric phase offsets. For more details about Gabor filters and the use of their parameters (aspect ratio, the standard deviation of the Gaussian envelope, etc.), we refer the reader to (Petkov, 1995; Petkov and Kruizinga, 1997; Kruizinga and Petkov, 1999; Grigorescu et al., 2002; Petkov and Westenberg, 2003; Grigorescu et al., 2003a). We normalise the Gabor functions that we use so all positive values sum up to 1 whereas all negative values sum up to -1. In this way, the response to an image of constant in-
tensity is always 0 and the largest response to a line of width \( w \) is achieved using a symmetrical filter with \( \lambda = 2w \).

The response of a Gabor filter is computed by convolving the input image with a Gabor kernel of preferred parameter values. We obtain a new kernel from each given Gabor kernel that we use. For symmetric filters, the new kernel is made up from the central part of the Gabor kernel whereas for antisymmetric filters it is made up from the largest positive part of the Gabor kernel. We denote by \( K_{\lambda,\theta,\zeta} \) such a kernel associated with its corresponding Gabor response \( g_{\lambda,\theta,\zeta,c}(x,y) \), note that the same kernel is used for every colour channel.

In order to detect lines or edges, we compute the L-infinity norm of the three Gabor responses obtained for each colour channel.

\[
g_{\lambda,\theta,\zeta,c}(x,y) = \max_{z=1,2,3} \{ g_{\lambda,\theta,\zeta,c_z}(x,y) \} \tag{5.2}
\]

Then, we compute the L-infinity norm across the two values of \( \zeta \) used. We use \( \zeta = \{0, \pi\} \) for line detection and \( \zeta = \{\pi/2, 3\pi/2\} \) for edge detection. We analyse both values of \( \zeta \) to achieve independence from the background luminance.

\[
g_{\lambda,\theta}(x,y) = \max_{z=1,2} \{ g_{\lambda,\theta,\zeta_z,c}(x,y) \} \tag{5.3}
\]

Finally, we threshold the responses of Gabor filters at a fraction \( t_1 (0 \leq t_1 \leq 1) \) of the maximum response of \( g_{\lambda,\theta}(x,y) \) across all combinations of values \( (\lambda, \theta) \) used and all positions \( (x,y) \) in the image, and denote these thresholded responses as \( |g_{\lambda,\theta}(x,y)|_{t_1} \). This operation rejects low responses of Gabor filters that fall under a local threshold.

**Contour part and colour description.** The colour COSFIRE filter is configured around a selected point of interest, which we consider as the centre of the filter. This point can be either manually selected by an user or automatically set as the central pixel of the ROI. We take the responses of a bank of Gabor filters, characterised by parameter values \( (\lambda, \theta) \), along circumferences of given radii \( \rho \) around the point of interest, Fig. 5.7. When \( \rho = 0 \), we only consider the point of interest. The colour COSFIRE filter is defined at certain positions \( (\rho_i, \phi_i) \) with respect to the point of interest in which there are local maxima responses of the bank of Gabor filters. A set of seven parameter values \( (\lambda_i, \theta_i, \rho_i, \phi_i, \gamma_{1i}, \gamma_{2i}, \gamma_{3i}) \) characterizes the properties of a contour part that is present in the specified pattern of interest: \( \lambda_i/2 \) represents the width, \( \theta_i \) represents the orientation, \( (\rho_i, \phi_i) \) represents the location and \( (\gamma_{1i}, \gamma_{2i}, \gamma_{3i}) \) represents the colour description at each colour channel. In the following, we explain how we obtain the parameter values of such contour parts.

First, we consider the responses of a bank of Gabor filters, \( |g_{\lambda,\theta}(x,y)|_{t_1} \), along a circumference of radius \( \rho \) around the point of interest. In each position along that circumference, we take the maximum of all responses across the possible values
5.2. Adding colour description to COSFIRE filters

Figure 5.7: Configuration of a colour COSFIRE filter. (a) The gray-level of a pixel represents the maximum value superposition of the thresholded (at $t_1 = 0.4$) responses of a bank of symmetric Gabor filters (4 wavelengths $\lambda \in \{3, 6, 10, 14\}$, 6 orientations $\theta = \{\frac{\pi}{6}, i = 0...5\}$ and $\zeta = 0$) at that position. The red cross indicates the location of the point of interest (in this case selected by the user) and the yellow circle represents the locations considered around the point of interest for a given radius $\rho$, here $\rho = 10$. (b) Values of the maximum value superposition of the thresholded responses of the bank of Gabor filters along the concerned circle. The three local maxima in the plot are respectively labelled and marked with black dots in (a). The local positions of the local maxima in (a) relative to the centre of the filter ($\rho_i, \phi_i$) and the wavelength and orientation ($\lambda_i, \theta_i$) of the Gabor filter that produced such response describe partly a tuple.

For such a location ($\rho_i, \phi_i$), we consider all combinations of ($\lambda, \theta$) for which the corresponding responses $|g_{\lambda,\theta}(x,y)|_{t_1}$ are greater than a fraction $t_2 = 0.75$ of the maximum of $|g_{\lambda,\theta}(x,y)|_{t_1}$ across the different combinations of values ($\lambda, \theta$) used. For further comment on the choice of the value of $t_2$, we refer the reader to [Azzopardi and Petkov, 2013c]). For each value $\theta$ that satisfies the previous condition, we consider a single value of $\lambda$, the one for which the corresponding response is the maximum of all responses across all values of $\lambda$. Each of the previous pairs ($\lambda, \theta$) in the location ($\rho_i, \phi_i$) describe partly a tuple ($\rho_i, \phi_i, \lambda_i, \theta_i$).

As for the colour description of the tuples, we compute the average of the pixel values in a region around the location ($\rho_i, \phi_i$) for each colour channel. We centre the kernel $K_{\lambda_i,\theta_i,\zeta}$ around the location ($\rho_i, \phi_i$) and perform a pixel-wise multiplication of the kernel by a colour channel of the prototype image $P_c$. Then, we normalise the result. Thus, we obtain a colour description value for each colour channel at the considered location, $\gamma_{c_i}$. 
5. Object recognition for content-based image retrieval

Figure 5.8: Regions of the prototype pattern, Fig. 5.6b, considered to compute the colour description in each contour part (white pixels are not considered). On the one hand, (a) and (c) correspond to the contour parts in the centre of the prototype. On the other hand, (b) corresponds to the labelled point 'a', (d) to 'b' and (e) to 'c' in Fig. 5.7a.

\[ \gamma_{c_i} = \frac{\sum_{k=1}^{m} \sum_{l=1}^{n} P_c(x_i + k - 1, y_i + l - 1) K_{\lambda_i, \theta_i, \zeta}(k, l)}{\sum_{k=1}^{m} \sum_{l=1}^{n} K_{\lambda_i, \theta_i, \zeta}(k, l)} \]  

(5.4)

where \( m \) and \( n \) are the rows and columns of the kernel \( K_{\lambda_i, \theta_i, \zeta} \) respectively and \((x_i, y_i)\) the Cartesian coordinates of \((\rho_i, \phi_i)\). We compute this average rather than directly using the value of the pixel \((\rho_i, \phi_i)\) at each colour channel to avoid that possible noisy values of pixels may deeply affect the colour description. Figure 5.8 shows the regions of the prototype pattern considered to compute the colour descriptions. For symmetric Gabor filters, both kernels \( K_{\lambda_i, \theta_i, \zeta} \) for values of \( \zeta \in \{0, \pi\} \) are identical, so anyone can be taken to describe the colour. Since we are using the central part of a symmetric Gabor filter of wavelength \( \lambda_i \), we ensure that the colour description is computed at a region of width equals to, at most, \( \lambda_i/2 \), which is the width of the line that the method localises. For antisymmetric Gabor filters, we use the kernel \( K_{\lambda_i, \theta_i, \zeta} \) with the value of \( \zeta \in \{\pi/2, 3\pi/2\} \) in which the Euclidean distance from the centroid of the kernel to the interest point is minimum when both kernels are centred around the location \((\rho_i, \phi_i)\). In this way, we describe the part of the prototype that is closer to the centre of the colour COSFIRE filter.

A set of seven parameter values or tuple \( p_i = (\lambda_i, \theta_i, \rho_i, \phi_i, \gamma_1, \gamma_2, \gamma_3) \) specifies the properties of a contour part. The set \( S_f = \{p_i|i = 1 \ldots n_c\} = \{(\lambda_i, \theta_i, \rho_i, \phi_i, \gamma_{1i}, \gamma_{2i}, \gamma_{3i})| i = 1 \ldots n_c\} \) denotes the parameter values combinations which fulfil the above conditions. The subscript \( f \) stands for the prototype pattern around the selected point of interest and \( n_c \) is the number of localised contour parts.

For the prototype shown in Fig. 5.6b and Fig. 5.7a, with two values of the parameter \( \rho \) \((\rho = \{0, 10\})\), this method results in five contour parts with parameter values specified by the tuples in the set shown in Table 5.3. The second tuple \((\lambda_2 = 10, \theta_2 = \pi/2, \rho_2 = 10, \phi_2 = 0, \gamma_{12} = 0, \gamma_{22} = 0, \gamma_{32} = 1)\) describes a contour part with a width of \((\lambda_2/2 =)\) 5 pixels and an orientation of \( \theta_2 = \pi/2 \) that can
Table 5.3: Set of tuples that describe the contour parts of the prototype in Fig. 5.6b and 5.7a.

\[ S_f = \{ \]
\[ (\lambda_1 = 10, \theta_1 = \pi/2, \rho_1 = 0, \phi_1 = 0, \gamma_{1_1} = 0.2, \gamma_{2_1} = 0, \gamma_{3_1} = 0.6), \]
\[ (\lambda_2 = 10, \theta_2 = \pi/2, \rho_2 = 10, \phi_2 = 0, \gamma_{1_2} = 0, \gamma_{2_2} = 0, \gamma_{3_2} = 1), \]
\[ (\lambda_3 = 6, \theta_3 = 0, \rho_3 = 0, \phi_3 = 0, \gamma_{1_3} = 1, \gamma_{2_3} = 0, \gamma_{3_3} = 1), \]
\[ (\lambda_4 = 6, \theta_4 = 0, \rho_4 = 10, \phi_4 = \pi/2, \gamma_{1_4} = 1, \gamma_{2_4} = 0, \gamma_{3_4} = 1), \]
\[ (\lambda_5 = 6, \theta_5 = 0, \rho_5 = 10, \phi_5 = 3\pi/2, \gamma_{1_5} = 1, \gamma_{2_5} = 0, \gamma_{3_5} = 1) \}

Figure 5.9: Structure of the colour COSFIRE filter for the prototype in Fig. 5.6b. Each of the numbered ellipses represent a tuple of the set of contour parts shown in Table 5.3 labelled with the same identification numbers. The wavelengths and orientations of the Gabor filters at the local positions of the contour parts and the colours described for each contour part are taken into account for the representation. This filter is trained to detect the spatial local arrangement and colour of five contour parts. The bright blobs are intensity maps of the Gaussian functions that will be used in the application step for blurring the responses of the Gabor filters.

Application of a colour COSFIRE filter for vertex localisation

To obtain the response for line detection, we apply a bank of Gabor filters to an input image with the pairs of values \((\lambda, \theta)\) that form the tuples of the set \(S_f\). To compute the responses for colour description, we apply Gaussian kernels to measure the similarity between the colour descriptions of the configuration and the ones
from the input image. For each pixel, we obtain the responses at the local positions \((\rho_i, \phi_i)\) of \(S_f\) from the considered pixel in terms of lines detection and colour description. Since we want to achieve strong responses both for line detection and for color description for each contour part, we multiply the two responses. The output of the colour COSFIRE filter for each pixel in the image can be computed as a combination of all responses for the different contour parts defined in the configuration step. The concerned responses for each contour part are in different positions \((\rho_i, \phi_i)\) with respect to the filter centre, thus we first shift them appropriately so that they come together to the filter centre. In the following, we explain in detail these steps.

**Line/edge detection.** We compute the responses of a bank of 2D Gabor filters applied to each colour channel of the input image for the pairs of values \((\lambda_i, \theta_i)\) of the set \(S_f\) and for both phase offset values \(\zeta\). If symmetric Gabor filters were used in the configuration selection, \(\zeta = \{0, \pi\}\), otherwise if antisymmetric filters were applied, \(\zeta = \{\pi/2, 3\pi/2\}\). Both values of \(\zeta\) are analysed because we want the method to localise the pattern of interest independently of the background. In the same way as for the configuration of the colour COSFIRE filter, we apply two consecutive L-infinity norms, along the colour channels and along the phase offset values. Then we threshold the responses at a fraction \(t_1\) of the maximum response, resulting in a Gabor response \(|g_{\lambda_i, \theta_i}(x,y)|_{t_1}\) for each tuple \(p_i\) in the set \(S_f\). We also obtain the Kernels \(K_{\lambda_i, \theta_i, \zeta}\) associated with the Gabor filters of each tuple.

The Gabor filter responses are blurred to allow for some tolerance in the position of the contour parts. We define the blurring operation as a convolution, both along the rows and columns, of the thresholded Gabor responses \(|g_{\lambda_i, \theta_i}(x,y)|_{t_1}\) with a rotationally symmetric Gaussian lowpass filter \(G_\sigma(x,y)\) of size \(1 \times n\sigma\) pixels with standard deviation \(\sigma\). The standard deviation is a linear function of the distance \(\rho\) from the centre of the colour COSFIRE filter,

\[
\sigma = \sigma_0 + \alpha \rho \quad (5.5)
\]

where \(n, \sigma_0\) and \(\alpha\) are constants. The orientation bandwidth is broader with a higher value of \(\alpha\). The visual system of the brain inspired the choice of the linear function in Eq. 5.5 following [Azzopardi and Petkov 2013c]. The blurred response for a tuple \(p_i\) is

\[
b_{\lambda_i, \theta_i, \sigma}(x,y) = |g_{\lambda_i, \theta_i}(x,y)|_{t_1} * G_\sigma(x,y) * G'_\sigma(x,y) \quad (5.6)
\]

Next, we shift the blurred responses of each tuple \(p_i\) by a distance of \(\rho_i\) in the opposite direction to \(\phi_i\). In polar coordinates, we can express this as \(\rho_i \phi_i + \pi\), whereas in cartesian coordinates it is described as an increment \((\Delta x_i, \Delta y_i)\) where \(\Delta x_i = -\rho_i \cos \phi_i\) and \(\Delta y_i = -\rho_i \sin \phi_i\). We denote by \(s_{\lambda_i, \theta_i, \sigma, \phi_i}(x,y)\) the blurred and shifted response of the Gabor filter specified by the tuple \(p_i\) in the set \(S_f\):
5.2. Adding colour description to COSFIRE filters

\[ s_{\lambda_i, \theta_i, \rho_i}(x, y) = b_{\lambda_i, \theta_i, \rho_i}(x - \Delta x_i, y - \Delta y_i) \quad (5.7) \]

Figure 5.10 shows the application of a colour COSFIRE filter to an input image for line detection. The response of the colour COSFIRE filter for line detection is achieved by computing five blurred and shifted responses of two pairs of Gabor filters. Each pair of Gabor filters has the same parameters except for their phase offset values, which are complementary, and they are both combined by taking the maximum value of the response for every pixel. Each of the five responses corresponds to each contour part found in the configuration.

**Colour description.** First, for each tuple \( p_i \), we convolve each colour channel of the input image \( I_c \) with the corresponding sliding kernels \( K_{\lambda_i, \theta_i, \zeta} \) and then we normalise the results.

\[ v_{\lambda_i, \theta_i, c}(x, y) = I_c(x, y) \ast K_{\lambda_i, \theta_i, \zeta}(x, y) \sum_{k=1}^{m} \sum_{l=1}^{n} K_{\lambda_i, \theta_i, \zeta}(k, l) \quad (5.8) \]

where \( k \) and \( l \) are the rows and columns of the kernel \( K_{\lambda_i, \theta_i, \zeta} \) respectively.

For symmetric Gabor filters, since the same kernel is computed for both values of \( \zeta \), any of them can be used as the sliding kernel. For antisymmetric Gabor filters, we compute the convolutions with \( K_{\lambda_i, \theta_i, \zeta=\pi/2} \) and \( K_{\lambda_i, \theta_i, \zeta=3\pi/2} \).

We denote by \( h_{p_i}(x, y) \) the response for colour description of the tuple \( p_i \) in the set \( S_f \). We compute \( h_{p_i}(x, y) \) by applying a Gaussian kernel that measures the similarity between the colours of the prototype contour part and the colours of the input image in each colour channel as in Eq. 5.9.

\[ h_{p_i}(x, y) = \exp \left( -\sum_{c=1}^{3} \frac{(v_{\lambda_i, \theta_i, c}(x, y) - \gamma_c)^2}{2\sigma_g^2} \right) \quad (5.9) \]

where \( \sigma_g \) is the standard deviation of the colour Gaussian kernel.

For antisymmetric Gabor filters, we compute a response for colour description \( h'_{p_i}(x, y) \) for each Gabor kernel and then we obtain the maximum value along both responses for each pair of corresponding pixels \( (x_j, y_k) \).

\[ h'_{p_i}(x, y) = \max_{j,k} \{ h_{p_i}(x_j, y_k) | \zeta = \pi/2, h_{p_i}(x_j, y_k) | \zeta = 3\pi/2 \} \quad (5.10) \]

Afterwards, we blur the response for colour description.

\[ h''_{p_i}(x, y) = h'_{p_i}(x, y) \ast G_{\sigma_i}(x, y) \ast G'_{\sigma_i}(x, y) \quad (5.11) \]

And finally, we shift the blurred response for colour description a distance of \( \rho_i \) in the opposite direction to \( \phi_i \).

\[ \hat{h}_{p_i}(x, y) = h''_{p_i}(x - \Delta x_i, y - \Delta y_i) \quad (5.12) \]
### 5. Object recognition for content-based image retrieval

#### Prototype Structure

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

#### Input image

- \( \lambda_1 = \lambda_2 = 10 \)
- \( \theta_1 = \theta_2 = \pi/2 \)
- \( \zeta = 0 \)

#### Gabor filters

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>i = 1, ( \zeta = 0 )</td>
<td>i = 2, ( \zeta = 0 )</td>
<td>i = 3, ( \zeta = \pi/2 )</td>
<td>i = 4, ( \zeta = 0 )</td>
<td>i = 5, ( \zeta = \pi/2 )</td>
</tr>
</tbody>
</table>

#### Gabor responses

- \( g_{\lambda_1, \theta_1, \zeta}(x, y) \)
- \( g_{\lambda_2, \theta_2, \zeta}(x, y) \)

#### Blurred and shifted Gabor responses

- \( s_{\lambda_1, \theta_1, \rho_1, \phi_1}(x, y) \)
- \( s_{\lambda_2, \theta_2, \rho_2, \phi_2}(x, y) \)

**Figure 5.10:** The ‘×’ marker indicates the location of the point of interest. (a) Input image. The framed area shows (left) the enlarged pattern of interest selected for the configuration and (right) the structure of the colour COSFIRE filter that was configured for this pattern. (b) Each contour part of the prototype pattern is detected by the combination of the responses of a pair of symmetric Gabor filters with preferred values of wavelength \( \lambda_i \) and orientation \( \theta_i \) and phase offsets \( \zeta_i = \{0, \pi\} \). Two of the contour parts \( (i = \{1, 2\}) \) are detected by one pair of Gabor filters and the other three parts \( (i = \{1, 2, 3\}) \) are detected by another pair of Gabor filters. Thus, only two pairs of distinct Gabor filters are chosen from the filter bank. (c) The thresholded response \( |g_{\lambda_1, \theta_1}(x, y)|_{t_1} \) (here \( t_1 = 0.2 \)) is then blurred (here \( n = 6 \)) and later shifted by \( (\rho_i, \phi_i + \pi) \) in polar coordinates.

**Figure 5.11** shows the application of a colour COSFIRE filter to an input image for colour description. The response of the colour COSFIRE filter for colour description is achieved by computing five blurred and shifted responses of three
5.2. Adding colour description to COSFIRE filters

Gaussian kernel similarities. Each of the five responses corresponds to each contour part found in the configuration.

Figure 5.11: The ‘×’ marker indicates the location of the point of interest. (a) Input image. The framed area shows (left) the enlarged pattern of interest selected for the configuration and (right) the structure of the colour COSFIRE filter that was configured for this pattern. (b) The image is convolved with two sliding kernels defined by the two Gabor filters used for line detection and then is normalised. (c) We compute the Gaussian kernel similarity between the colours of the prototype contour part and the colours of the input image. There are only three tuples with unique values of \((\lambda_i, \theta_i, \gamma_1, \gamma_2, \gamma_3)\) and therefore only three similarities are obtained. (d) The Gaussian kernel responses are then blurred (here \(n = 6\)) and later shifted by \((\rho, \phi + \pi)\) in polar coordinates.

Response of a colour COSFIRE filter. We define the response of a colour COSFIRE filter \(r_{S_f}(x, y)\) as the weighted geometric mean of the Hadamard product of
the blurred and shifted thresholded Gabor filter responses, \( s_{\lambda_i, \theta_i, \rho_i, \phi_i}(x, y) \), by the blurred and shifted Gaussian colour responses, \( \hat{h}_{p_i}(x, y) \), that correspond to the properties of the contour parts described in \( S_f \):

\[
\begin{align*}
  r_{S_f}(x, y) & \overset{\text{def}}{=} \left( \prod_{i=1}^{|S_f|} (s_{\lambda_i, \theta_i, \rho_i, \phi_i}(x, y) \circ \hat{h}_{p_i}(x, y)) \right)^{1/\sum_{i=1}^{|S_f|} \omega_i} \\
  \omega_i & = \exp^{-\frac{\rho_i^2}{2\sigma_i'}} \\
  \sigma_i' & = \left(-\rho_{\text{max}}^2/2\ln\tau\right)^{1/2} \\
  \rho_{\text{max}} & = \max_{i \in 1...|S_f|} \{\rho_i\}
\end{align*}
\]

where \( s_{\lambda_i, \theta_i, \rho_i, \phi_i}(x, y) \circ \hat{h}_{p_i}(x, y) \) stands for the Hadamard product of \( s_{\lambda_i, \theta_i, \rho_i, \phi_i}(x, y) \) and \( \hat{h}_{p_i}(x, y) \). When \( 1/\sigma_i' = 0 \), the weighted geometric mean becomes a standard geometric mean and all the contour parts responses \( s_{\lambda_i, \theta_i, \rho_i, \phi_i}(x, y) \circ \hat{h}_{p_i}(x, y) \) have the same contribution. On the contrary, for \( 1/\sigma_i' > 0 \) the contribution of the contour parts decreases with an increasing value of the corresponding parameter \( \rho \). In particular, we achieve a maximum value \( \omega = 1 \) of the weights in the centre (\( \rho = 0 \)), and minimum value \( \omega = \tau \) in the periphery (\( \rho = \rho_{\text{max}} \)).

Finally, we threshold the response of the colour COSFIRE filter at a fraction \( t_3 \) of its maximum across all image coordinates \( (x, y) \), \( 0 \leq t_3 \leq 1 \).

\[
r(x, y) = |r_{S_f}(x, y)|_{t_3}
\]

Figure 5.12 shows the application of a colour COSFIRE filter to an input image for the localisation of a colour vertex. The output of the colour COSFIRE filter is the weighted geometric mean of the Hadamard multiplication of five blurred and shifted responses of two pairs of Gabor filters and five blurred and shifted responses of three convolutions. The filter responds at points where there is a pattern identical or similar to the prototype pattern (Fig. 5.6b) and at the point of interest of the prototype pattern despite the different colors and patterns of the background. Thus, we are getting strong responses in a given point to a local pattern that contains a horizontal blue line to the right of the aforementioned point, a vertical pink line above and under the point and a horizontal bluish and a vertical pink lines at the point.
5.2. Adding colour description to COSFIRE filters

Figure 5.12: The ‘×’ marker indicates the location of the point of interest. (a) Blurred and shifted Gabor responses $s_{\lambda_{i}, \theta_{i}, \rho_{i}, \phi_{i}}(x, y)$. (b) Colour shifted responses $h_{p_{i}}(x, y)$. (c) Contour part shifted responses $s_{\lambda_{i}, \theta_{i}, \rho_{i}, \phi_{i}}(x, y) \circ h_{p_{i}}(x, y)$. (d) COSFIRE colour output $r_{S_{j}}$.

\[
\left( \prod_{j=1}^{5} (s_{\lambda_{i}, \theta_{i}, \rho_{i}, \phi_{i}}(x, y) \circ h_{p_{i}}(x, y))^\omega_j \right)^{1/\Omega}
\]

where
\[
\begin{align*}
\omega_1 &= 1, \omega_2 = 0.5, \omega_3 = 1, \omega_4 = 0.5, \omega_5 = 0.5 \\
\Omega &= \omega_1 + \omega_2 + \omega_3 + \omega_4 + \omega_5 = 3.5
\end{align*}
\]

5.2.2. Method with application for colour object localisation

Overview

The previous method describes the colour of lines or edges but the colour of an object is also defined by the colour of its surface. We define a new set of tuples for the colour description of blobs in the surface of a prototypical object of interest. For each contour part of the new set of tuples, we compute the response of the colour description of blobs in an input image in the same way as the response for colour
5. Object recognition for content-based image retrieval

Figure 5.13: 5.13a Synthetic input image of size 700 × 550 pixels. 5.13b Enlargement of the prototype from which the colour COSFIRE filter will be obtained. It corresponds to the top left object in 5.13a. The white cross indicates the centre of the prototype, in this case automatically assigned as the centre of the ROI.

description of lines. The response of the colour COSFIRE filter is obtained by the Hadamard multiplication of the response for colour edge detection, explained in the previous section, and the weighted geometric mean of the blurred and shifted Gaussian kernel similarities for colour description of blobs.

Figure 5.13a shows an input image with four objects. We consider the top left object as the prototype of interest. The ROI that encompass the prototype is shown enlarged in Fig. 5.13b. The colour COSFIRE filter configured for this prototype will respond to the same and similar patterns in terms of shape and colours.

In the succeeding sections we explain the extraction of the new set of tuples for colour description of blobs, the extraction of the response of the colour description of blobs for an input image and the computation of the colour COSFIRE filter response.

Configuration of a colour COSFIRE filter for object localisation

We use SIFT detector (Lowe 2004) to look for stable keypoints in the prototype. SIFT is a blob detection method that localises regions in images that differ in properties compared to the surrounding regions. A SIFT keypoint is a circular image region described by a geometric frame of four parameters: the keypoint centre coordinates \((x_j, y_j)\), its scale (that it is equal to the radius of the region) \(\delta_j\), and its orientation, (Vedaldi and Fulkerson 2008). We are only interested in the coordinates and scale of the keypoints.

We apply SIFT detector to every channel of the input image \(I_c\) and consider the keypoints whose scale is greater than a fraction \(t_4\) of the maximum scale across all keypoints. Then we cluster the remaining keypoints into three groups according to their scale values using \(k\)-means (Duda et al. 2000), and assign to each keypoint the
5.2. Adding colour description to COSFIRE filters

Figure 5.14: The green circles show the keypoints found using SIFT detector for the prototype object of interest. (a) All unique SIFT keypoints detected. The radius of the circle represents the scale at which the keypoint was found. (b) Remaining keypoints after thresholding with $t_4 = 0.2$. (c) Keypoints with only three scales.

The point of interest of the prototype $(x_p, y_p)$, which is the centre of the colour COSFIRE filter, can be either manually selected by the user or automatically assigned as the centre of the ROI. We compute the local polar coordinates $(\rho_j, \phi_j)$ of the keypoints $(x_j, y_j)$ with respect to the point of interest of the prototype pattern.

\[
(\rho_j, \phi_j) = \left( \sqrt{(x_j - x_p)^2 + (y_j - y_p)^2}, \text{atan2}(y_j - y_p, x_j - x_p) \right)
\]

(5.18)

where $\text{atan2}$ is the angle in radians between the positive x-axis of a plane and the point given by the coordinates $(x_j, y_j)$ on it.

For each keypoint $(\delta_j, \rho_j, \phi_j)$, we create a Gaussian circular mask $K_{\delta_j, \rho_j, \phi_j}(x, y)$ of radius $\delta_j$ centred at the corresponding locations $(\rho_j, \phi_j)$.

\[
K_{\delta_j, \rho_j, \phi_j}(x, y) = \exp\left(-\frac{x^2 + y^2}{2\delta_j^2}\right)
\]

(5.19)

We then perform a pixel-wise multiplication of the mask by each colour channel of the prototype $P_c$ and then normalise the result. In this way, the pixels closer to the considered location have a stronger participation in the computation of the colour description of the blob than the ones at further distances, Fig. 5.15. Therefore, we obtain a colour description value for each colour channel $\gamma_{cj}$ at the considered keypoint $(\delta_j, \rho_j, \phi_j)$. 

mean scale value of the group to which they belong. This step is not essential but it allows to speed up later computations. Finally, only unique keypoints $(\delta_j, x_j, y_j)$ are kept, Fig. 5.14.
5. Object recognition for content-based image retrieval

Figure 5.15: Colour description of a blob. (a) Prototypical object of interest. SIFT keypoints are marked in green. We choose one keypoint, marked in blue, as example for the computation of the colour description of that blob. (b) Gaussian circular mask \( K_{\delta_j,\phi_j} (x,y) \) for the keypoint marked in blue. (c) Result of the pixel-wise multiplication of the Gaussian circular mask by the prototypical object of interest. The colour description of this keypoint for RGB colour space results in \( \gamma_{c_1} = 0, \gamma_{c_2} = 1, \gamma_{c_3} = 1 \) = [0, 1, 1] which is cyan colour.

\[
\gamma_{c_j} = \frac{\sum_{k=1}^{m} \sum_{l=1}^{n} P_c(x_j + k - 1, y_j + l - 1) K_{\delta_j, \phi_j}(x, y) \gamma_{c_j}}{\sum_{k=1}^{m} \sum_{l=1}^{n} K_{\delta_j, \phi_j}(x, y)} \quad (5.20)
\]

where \( m \) and \( n \) are the rows and columns of the kernel \( K_{\delta_j, \phi_j} \) respectively and \( (x_j, y_j) \) the Cartesian coordinates of \( (\rho_j, \phi_j) \).

A set of six parameter values or tuple \( p_j = (\delta_j, \rho_j, \phi_j, \gamma_{c_1}, \gamma_{c_2}, \gamma_{c_3}) \) specifies the properties of a contour part in this new set \( S'_f = \{ p_j | j = 1 \ldots n_k \} = \{ (\delta_j, \rho_j, \phi_j, \gamma_{c_1}, \gamma_{c_2}, \gamma_{c_3}) | j = 1 \ldots n_k \} \). The subscript \( f \) stands for the prototype object of interest around the point of interest and \( n_k \) is the number of detected keypoints.

We compute another set of tuples \( S_f = \{ p_i | i = 1 \ldots n_c \} = \{ (\lambda_i, \theta_i, \rho_i, \phi_i, \gamma_{c_1}, \gamma_{c_2}, \gamma_{c_3}) | i = 1 \ldots n_c \} \) for the object of interest as in Section 5.2.1.2 using a bank of antisymmetric Gabor filters with \( \lambda = 20 \) and \( \theta = \{ 0, \pi/6, \pi/3, \pi/2, 2\pi/3, 5\pi/6 \} \).

For the prototype shown in Fig. 5.15, this method results in two sets of tuples. Regarding the colour edge description, we localise 67 contour parts in a set \( S_f \). As for the colour description of blobs with \( t_4 = 0.2 \), we localise 12 contour parts or keypoints in a set \( S'_f \). Table 5.4 indicates the parameter values for three of those 12 keypoints. The third tuple describes a keypoint with a scale of \( \delta_3 = 28.7 \) pixels that can be detected by a SIFT detector at position \( \rho_3 = 120.9 \) pixels to the top right-hand corner \( (\phi_3 = \pi/4) \) of the point of interest (centre of the ROI) and with RGB colour description \( [\gamma_{c_1} = 0, \gamma_{c_2} = 1, \gamma_{c_3} = 1] = [0, 1, 1] \) which is cyan. This selection is the result of the presence of a cyan colour blob to the top right-hand corner of the centre of the prototype that is used for the configuration of the filter. This structure is represented in Fig. 5.16.
5.2. Adding colour description to COSFIRE filters

Table 5.4: Three tuples that give examples of the colour description of blobs of the prototypical object of interest in Fig. 5.6b and 5.7a. A total of 12 tuples were automatically described.

\[ S_f = \{ \]
\( (\delta_1 = 56.8, \ \rho_1 = 0, \ \phi_1 = 0, \ \gamma_{11} = 1, \ \gamma_{21} = 0, \ \gamma_{31} = 0), \]
\( (\delta_2 = 28.7, \ \rho_2 = 120.9, \ \phi_2 = -3\pi/4, \ \gamma_{12} = 1, \ \gamma_{22} = 1, \ \gamma_{32} = 0), \]
\( (\delta_3 = 28.7, \ \rho_3 = 115.3, \ \phi_3 = \pi/4, \ \gamma_{13} = 0, \ \gamma_{23} = 1, \ \gamma_{33} = 1), \}

Figure 5.16: Structure of the colour COSFIRE filter for colour description of blobs of the prototypical object in Fig. 5.13b. Each of the numbered circles represent a tuple of the set of contour parts shown in Table 5.4 with the same identification numbers. The wavelengths and orientations of the Gabor filters at the local positions of the contour parts and the colours described for each contour part are taken into account for the representation of the ellipses. The bright blobs are intensity maps of the Gaussian functions that are used in the application step for blurring the responses of the Gabor filters. The scale and colour described for each contour part for the colour description of blobs are considered for the representation of the circles. This filter is trained to detect the spatial local arrangement and colour of two sets of contour parts, one for colour edges and another for colour blobs.

Application of a colour COSFIRE filter for object localisation

We obtain the response for colour description of blobs by applying Gaussian kernels to measure the similarity between the colour descriptions of blobs at the configuration and the ones of the input image. Thus, this computation shares the main steps with the one used for the colour evaluation of lines and edges. The output of the colour COSFIRE filter is computed as the Hadamard product of the output obtained in [5.2.1.3] for colour edge detection and the response for colour description of blobs.

Colour description of blobs. For each unique value of \( \delta_j \) in the tuples of \( S'_f \), we compute a Gaussian circular mask \( K_{\delta_j}(x, y) \) that contains a circle of radius \( \delta_j \) defined as in Eq. 5.19. Then we convolve each colour channel of the input image \( I_c \) with the mask \( K_{\delta_j}(x, y) \) and normalise the results, as in Eq. 5.8.
We denote by $d_{p_j}(x,y)$ the response for colour description of blobs for the tuple $p_j$ in the set $S'_f$. We compute $d_{p_j}(x,y)$ by applying a Gaussian kernel that measures the similarity between the colours of the contour part defined by the tuple $p_j$ and the colours of the corresponding normalised and convolved input image along each colour channel, as in Eq. 5.9.

Afterwards, we blur the colour response, Eq. 5.11, and shift the blurred colour response a distance of $\rho_j$ in the opposite direction to $\phi_j$, Eq. 5.12, obtaining $\hat{d}_{p_j}$.

Response of a colour COSFIRE filter. We define the output $r_{S'_f}(x,y)$ of a colour COSFIRE filter for colour description of blobs in an object of interest as the weighted geometric mean of the blurred and shifted Gaussian similarity responses $d_{p_j}(x,y)$ that correspond to the properties of the contour parts described in $S'_f$:

\[
r_{S'_f}(x,y) \overset{\text{def}}{=} \left( \prod_{j=1}^{||S'_f||} \left( \hat{d}_{p_j}(x,y) \right)^{\omega_j} \right)^{1/\sum_{j=1}^{||S'_f||} \omega_j}
\]

where $\omega_j$ is defined in Eq. 5.14.

We compute the response of the colour COSFIRE filter $r(x,y)$ as the thresholded Hadamard product of the responses for colour edge detection and for colour description of blobs:

\[
r(x,y) \overset{\text{def}}{=} \left| r_{S_f}(x,y) \circ r_{S'_f}(x,y) \right|_{t_5}
\]

where $||_t$ stands for thresholding the response at a fraction $t_5$ of its maximum across all image coordinates $(x,y)$.

Figure 5.17 shows the application of a colour COSFIRE filter for localisation of colour objects. The output of the colour COSFIRE filter is the Hadamard product of the weighted geometric mean of 12 responses for colour description of blobs and the weighted geometric mean of 67 responses for colour edges detection. The filter responds at points where there is an identical or similar pattern to the prototypical object of interest (Fig. 5.13b) and at the point of interest of the prototypical object of interest despite the different colors and patterns of the background. Thus, we are getting strong responses in a given point to a local pattern that contains a red square centred at the aforementioned point, a yellow circle at the bottom left-hand corner of the square and a cyan circle at the top right-hand corner of the square.

For the achievement of invariance to rotation, scale, reflection and contrast inversion of the colour COSFIRE filter, we refer the reader to (Azzopardi and Petkov, 2013c).
5.2. Adding colour description to COSFIRE filters

Figure 5.17: Application demonstration for localisation of colour objects. (a) Input image, prototype and structure of the colour COSFIRE filter. Numbers indicate three tuples in $S_f$ for which we illustrate this application. (b) Normalised convolution of the input image by a Gaussian circular mask of radius the scale of the contour part considered. (c) Similarity between the colours of the contour parts and the colours in the input image by a Gaussian kernel. (d) We blur and shift the previous responses. (e) The output of the colour COSFIRE filter for colour description of blobs is obtained as a weighted geometric mean of the blurred and shifted responses, $r_{S_f}$. (f) We compute the output of the colour COSFIRE filter as the Hadamard product of $r_{S_f}$ and the output for colour edges detection, $r_{S_f}$. The three local maxima in this output correspond to the three similar objects in the input image.
5. Object recognition for content-based image retrieval

5.2.3. Experiments and results

Dataset

We use COIL-100 public benchmark in our experiments. It consists of colour images of 100 object classes of size 128×128. 72 images of each object were taken which sums up to 7200 images for the whole dataset. The images were obtained by placing the objects on a turntable and taking a snapshot every 5°. The objects have a wide variety of complex geometric and pose characteristics. Images do not present occlusion, background clutter and illumination changes. Figure 5.18 shows the image taken at 0° for all objects of COIL-100 whereas Fig. 5.18 shows the viewpoints of three objects at 0°, 45°, 90°, 135°, 180°, 225°, 270° and 315°.

Experimental set up and results

We configure one colour COSFIRE filter per object class for the image with rotation angle of 0°. We also configure standard COSFIRE filters for the same images. We use the same parameters for both colour and standard COSFIRE filters. We created a bank of Gabor filters with wave length \( \lambda = 5 \), orientations \( \theta = \{0, \pi/8, ..., \pi - \pi/8\} \), phase offsets \( \zeta = \{\pi/2, 3\pi/2\} \) and aspect ratio 0.4. We set thresholds \( t_1 = 0.1 \), \( t_2 = 0.75 \) and \( t_3 = 0 \), and parameters related with the standard deviation of the blurring function equal to \( \sigma_0 = 0.83 \) and \( \alpha = 0.1 \). We obtain the output of a COSFIRE filter by the geometric mean, thus \( \omega = 1 \). For colour description, we set \( \sigma_g = 0.04 \).

Figure 5.20 shows examples of the structures of the colour COSFIRE filters. The structures of the standard COSFIRE filters present the exact same tuples for contour description but without colour information.

We apply each configured COSFIRE filter to the whole dataset and compute precision and recall at every position in the retrieved hit list. We calculate the average precision, AveP, which is the area under a precision-recall curve, as

\[
\text{AveP} = \frac{\sum_{k=1}^{n}(P@k \times \text{rel}[k])}{\text{number of relevant images}}
\]  

(5.23)

where \( k \) is the rank in the sequence of retrieved images, \( n \) is the number of retrieved images, \( P@k \) is the precision at cut \( k \) in the hit list and \( \text{rel}[k] \) is 1 if the \( k \)th hit in the hit list is relevant and 0 otherwise.

Figure 5.21 shows plots of some precision-recall curves both for colour and standard COSFIRE filters. Table 5.5 indicates the average precisions obtained for each object class with both colour and standard COSFIRE filters. Colour COSFIRE filters have a higher distinctiveness power than standard COSFIRE filters since they always obtained higher average precisions.
5.2. Adding colour description to COSFIRE filters

Figure 5.18: COIL dataset. Images taken at 0° of each object class. These are the objects considered for the configuration of colour COSFIRE filters.

We compute the mean average precision, MAP, for all the queries of the dataset as the mean of the average precision scores for each query,

\[
\text{MAP} = \frac{\sum_{q=1}^{Q} \text{AveP}(q)}{Q}
\]

(5.24)

where \(Q\) is the number of queries.

We also obtain the maximum harmonic mean of precision and recall for each query of the dataset. We compute the mean harmonic mean of precision and recall,
5. Object recognition for content-based image retrieval

Figure 5.19: Viewpoints of three objects of COIL dataset at 0°, 45°, 90°, 135°, 180°, 225°, 270° and 315°.

Table 5.5: Average precisions of the 100 classes, Obj, of COIL dataset for colour COSFIRE filters, C (for colour), and COSFIRE filters, G (for gray).

<table>
<thead>
<tr>
<th>Obj</th>
<th>C</th>
<th>G</th>
<th>Obj</th>
<th>C</th>
<th>G</th>
<th>Obj</th>
<th>C</th>
<th>G</th>
<th>Obj</th>
<th>C</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.81</td>
<td>0.13</td>
<td>26</td>
<td>0.92</td>
<td>0.16</td>
<td>51</td>
<td>0.47</td>
<td>0.27</td>
<td>76</td>
<td>1.00</td>
<td>0.10</td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>0.23</td>
<td>27</td>
<td>0.93</td>
<td>0.03</td>
<td>52</td>
<td>0.64</td>
<td>0.16</td>
<td>77</td>
<td>0.20</td>
<td>0.09</td>
</tr>
<tr>
<td>3</td>
<td>0.71</td>
<td>0.00</td>
<td>28</td>
<td>0.44</td>
<td>0.00</td>
<td>53</td>
<td>1.00</td>
<td>0.02</td>
<td>78</td>
<td>0.77</td>
<td>0.01</td>
</tr>
<tr>
<td>4</td>
<td>1.00</td>
<td>0.01</td>
<td>29</td>
<td>0.90</td>
<td>0.35</td>
<td>54</td>
<td>0.50</td>
<td>0.12</td>
<td>79</td>
<td>0.41</td>
<td>0.14</td>
</tr>
<tr>
<td>5</td>
<td>1.00</td>
<td>0.38</td>
<td>30</td>
<td>1.00</td>
<td>0.00</td>
<td>55</td>
<td>0.30</td>
<td>0.17</td>
<td>80</td>
<td>0.15</td>
<td>0.09</td>
</tr>
<tr>
<td>6</td>
<td>0.64</td>
<td>0.21</td>
<td>31</td>
<td>0.47</td>
<td>0.26</td>
<td>56</td>
<td>1.00</td>
<td>0.01</td>
<td>81</td>
<td>0.86</td>
<td>0.57</td>
</tr>
<tr>
<td>7</td>
<td>1.00</td>
<td>0.05</td>
<td>32</td>
<td>0.96</td>
<td>0.24</td>
<td>57</td>
<td>0.38</td>
<td>0.09</td>
<td>82</td>
<td>0.61</td>
<td>0.01</td>
</tr>
<tr>
<td>8</td>
<td>0.67</td>
<td>0.19</td>
<td>33</td>
<td>0.57</td>
<td>0.17</td>
<td>58</td>
<td>1.00</td>
<td>0.01</td>
<td>83</td>
<td>1.00</td>
<td>0.04</td>
</tr>
<tr>
<td>9</td>
<td>0.21</td>
<td>0.13</td>
<td>34</td>
<td>1.00</td>
<td>0.06</td>
<td>59</td>
<td>1.00</td>
<td>0.20</td>
<td>84</td>
<td>1.00</td>
<td>0.07</td>
</tr>
<tr>
<td>10</td>
<td>0.91</td>
<td>0.01</td>
<td>35</td>
<td>1.00</td>
<td>0.19</td>
<td>60</td>
<td>0.30</td>
<td>0.04</td>
<td>85</td>
<td>0.45</td>
<td>0.11</td>
</tr>
<tr>
<td>11</td>
<td>0.66</td>
<td>0.00</td>
<td>36</td>
<td>0.71</td>
<td>0.00</td>
<td>61</td>
<td>0.44</td>
<td>0.05</td>
<td>86</td>
<td>1.00</td>
<td>0.01</td>
</tr>
<tr>
<td>12</td>
<td>0.37</td>
<td>0.20</td>
<td>37</td>
<td>0.94</td>
<td>0.56</td>
<td>62</td>
<td>0.29</td>
<td>0.05</td>
<td>87</td>
<td>1.00</td>
<td>0.37</td>
</tr>
<tr>
<td>13</td>
<td>0.26</td>
<td>0.12</td>
<td>38</td>
<td>0.43</td>
<td>0.00</td>
<td>63</td>
<td>0.44</td>
<td>0.00</td>
<td>88</td>
<td>0.96</td>
<td>0.13</td>
</tr>
<tr>
<td>14</td>
<td>0.28</td>
<td>0.14</td>
<td>39</td>
<td>0.38</td>
<td>0.16</td>
<td>64</td>
<td>0.60</td>
<td>0.45</td>
<td>89</td>
<td>0.68</td>
<td>0.10</td>
</tr>
<tr>
<td>15</td>
<td>0.80</td>
<td>0.13</td>
<td>40</td>
<td>0.72</td>
<td>0.09</td>
<td>65</td>
<td>0.24</td>
<td>0.13</td>
<td>90</td>
<td>0.56</td>
<td>0.02</td>
</tr>
<tr>
<td>16</td>
<td>0.82</td>
<td>0.44</td>
<td>41</td>
<td>0.55</td>
<td>0.01</td>
<td>66</td>
<td>0.73</td>
<td>0.17</td>
<td>91</td>
<td>0.63</td>
<td>0.06</td>
</tr>
<tr>
<td>17</td>
<td>0.61</td>
<td>0.08</td>
<td>42</td>
<td>0.97</td>
<td>0.02</td>
<td>67</td>
<td>0.75</td>
<td>0.19</td>
<td>92</td>
<td>0.58</td>
<td>0.19</td>
</tr>
<tr>
<td>18</td>
<td>0.24</td>
<td>0.03</td>
<td>43</td>
<td>0.99</td>
<td>0.01</td>
<td>68</td>
<td>0.71</td>
<td>0.18</td>
<td>93</td>
<td>0.54</td>
<td>0.13</td>
</tr>
<tr>
<td>19</td>
<td>0.47</td>
<td>0.06</td>
<td>44</td>
<td>0.35</td>
<td>0.03</td>
<td>69</td>
<td>0.86</td>
<td>0.10</td>
<td>94</td>
<td>1.00</td>
<td>0.02</td>
</tr>
<tr>
<td>20</td>
<td>0.97</td>
<td>0.01</td>
<td>45</td>
<td>0.26</td>
<td>0.01</td>
<td>70</td>
<td>0.90</td>
<td>0.16</td>
<td>95</td>
<td>1.00</td>
<td>0.11</td>
</tr>
<tr>
<td>21</td>
<td>0.50</td>
<td>0.07</td>
<td>46</td>
<td>0.53</td>
<td>0.05</td>
<td>71</td>
<td>1.00</td>
<td>0.08</td>
<td>96</td>
<td>0.88</td>
<td>0.18</td>
</tr>
<tr>
<td>22</td>
<td>0.26</td>
<td>0.19</td>
<td>47</td>
<td>1.00</td>
<td>0.21</td>
<td>72</td>
<td>1.00</td>
<td>0.42</td>
<td>97</td>
<td>0.24</td>
<td>0.14</td>
</tr>
<tr>
<td>23</td>
<td>0.90</td>
<td>0.01</td>
<td>48</td>
<td>0.67</td>
<td>0.04</td>
<td>73</td>
<td>1.00</td>
<td>0.20</td>
<td>98</td>
<td>1.00</td>
<td>0.06</td>
</tr>
<tr>
<td>24</td>
<td>1.00</td>
<td>0.06</td>
<td>49</td>
<td>0.92</td>
<td>0.11</td>
<td>74</td>
<td>0.35</td>
<td>0.14</td>
<td>99</td>
<td>0.94</td>
<td>0.24</td>
</tr>
<tr>
<td>25</td>
<td>1.00</td>
<td>0.92</td>
<td>50</td>
<td>1.00</td>
<td>0.09</td>
<td>75</td>
<td>0.20</td>
<td>0.06</td>
<td>100</td>
<td>0.40</td>
<td>0.09</td>
</tr>
</tbody>
</table>
5.2. Adding colour description to COSFIRE filters

Figure 5.20: Structures of the colour COSFIRE filters for the first 20 classes of COIL dataset. The wavelengths and orientations of the Gabor filters at the local positions of the contour parts and the colours described for each contour part are taken into account for the representation of the ellipses. The bright blobs are intensity maps of the Gaussian functions that are used in the application step for blurring the responses of the Gabor filters. The scale and colour described for each contour part for the colour description of blobs are considered for the representation of the circles.
Figure 5.21: Precision-recall curves for the first 20 classes of COIL dataset. In green, precision-recall curves of the colour COSFIRE filters. In blue, precision-recall curves of the standard COSFIRE filters. Red diamonds indicate the maxima harmonic means of precision and recall.

Table 5.6 shows the values of MAP, MFScore, MPrecision and MRecall for CBIR demonstrating the effectiveness of colour COSFIRE filters with respect to standard MFScore, as the mean of the maxima harmonic means of precision and recall for all queries of the dataset. Mean precision, MPrecision, and mean recall, MRecall, are the means of the precisions and recalls, respectively, that obtained the maxima harmonic means for all queries of the dataset.
Table 5.6: Mean average precision, MAP; mean harmonic mean, MFScore; mean precision, MPrecision; and mean recall, MRecall, of COIL dataset for colour COSFIRE filters, C, and COSFIRE filters, G.

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>0.6970</td>
<td>0.1322</td>
</tr>
<tr>
<td>MFScore</td>
<td>0.7617</td>
<td>0.2241</td>
</tr>
<tr>
<td>MPrecision</td>
<td>0.9388</td>
<td>0.3217</td>
</tr>
<tr>
<td>MRecall</td>
<td>0.6822</td>
<td>0.3162</td>
</tr>
</tbody>
</table>

COSFIRE filters.

We also evaluate the performance of COSFIRE filters as a classification problem. The responses of a given COSFIRE filter are divided by the maximum response obtained with that filter. A given image is classified to the class by which the COSFIRE filter that achieves the maximum response was configured. We compute a confusion matrix where the value at location \((i, j)\) is the number of images of class \(i\) classified as class \(j\). Figures 5.22 and 5.23 show the confusion matrices of the colour COSFIRE filters and standard COSFIRE filters, respectively. The confusion matrix of the colour COSFIRE filters is less sparse than the one of the standard method, with high values at the diagonal and low values at the off-diagonal. The proposed colour-based approach yields 67.57% accuracy while the standard one achieves 21.69% accuracy, computing accuracy as the trace of the confusion matrix divided by the total number of images of the dataset.

5.3. Conclusions

The contributions of the work presented in this chapter are two-fold. First, we evaluated different clustering configurations of SIFT keypoints in relation with their pose parameters: coordinates location, scale and orientation. On the one hand, we used the similarity measure of the closest pairs of keypoint descriptors. On the other hand, we used a Hough transform, with different parametrization values, to identify clusters of at least three points voting for the same pose of an object and we verified the consistency of the pose parameters with the least squares algorithm. Second, we proposed colour COSFIRE filters that add colour description and discrimination to COSFIRE filters as well as providing invariance to background intensity. We presented colour COSFIRE filters both for patterns made up of colour lines and for patterns that are colour objects. Colour COSFIRE filters demonstrated to obtain higher retrieval and classification performance than the standard COSFIRE filters on COIL dataset.

All in all, in this section we contemplated two important tasks for object retrieval such as object matching and object localisation.
Figure 5.22: Confusion matrix for the proposed colour COSFIRE filters. The matrix is of size $100 \times 100$. The columns represent the instances in a predicted class and the rows the instances in an actual class.
Figure 5.23: Confusion matrix for the standard COSFIRE filters. The matrix is of size 100 × 100. The columns represent the instances in a predicted class and the rows the instances in an actual class.