Obtaining Feature Correspondences

Neill Campbell

May 9, 2008

A state-of-the-art system for finding objects in images has recently been developed by David Lowe. The algorithm is termed the Scale-Invariant Feature Transform (SIFT) and intends to detect similar feature points in each of the available images and then describe these points with a feature vector which is independent of image scale and orientation. Thus feature points which correspond to different views of the same object should have similar feature vectors. If this process is successful then we should be able to use a simple algorithm to compare the collected set of feature vectors from one image to another in order to find corresponding feature points in each image. Figure 1 provides a typical example of using SIFT to locate a query image within a search image.

The SIFT algorithm may be decomposed into four stages:

1. Feature point detection
2. Feature point localisation
3. Orientation assignment
4. Feature descriptor generation

We will now discuss the different stages. It should be noted that the terms feature point and keypoint are synonymous.

1 Feature Point Detection

Unlike feature detectors which use directed image gradients, such as the Harris corner detector, the SIFT algorithm uses a form of 'blob' detection. These detectors use the laplacian of an image, which contains no directional information, to evaluate isolated dark and light regions within the image. Convolution kernels, consisting of the Laplacian of Gaussian kernels of increasing variance, may be applied to the image to locate these dark or light regions at scales corresponding to the variance of the kernel. This allows detection of scale in addition to position within the image.

If we take an image $I(x, y)$ then we may apply a Gaussian filter of (1) to generate a smoothed image $S(x, y)$ at an appropriate variance σ^2 as given in (2).

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-(x^2 + y^2)}{2\sigma^2}\right) \tag{1}$$

$$S(x, y, \sigma) = g(x, y, \sigma) * I(x, y) \tag{2}$$

Figure 1: An example using SIFT to locate one image inside another. *The query image (left) and the search image (right) are processed using the SIFT algorithm to identify feature points and their corresponding feature descriptors. These descriptors can then be matched to locate corresponding points.*

We may locate the centres of dark and light regions in this smoothed image by finding the extreme values of the Laplacian of the image. If we normalise the Laplacian for the scale parameter σ then we may find extreme values in both position and scale to locate keypoints.

$$\nabla^2_{\text{norm}} S(x, y, \sigma) \quad = \quad \sigma^2 \left(S_{xx} + S_{yy}\right) \tag{3}$$

SIFT implements an efficient approximation of this Laplacian detector called the Difference of Gaussians (DoG) detector. This forms a Gaussian scale-space (GSS) representation of the image. The GSS is created by generating a series of smoothed images at discrete values of σ over a number of *octaves* where the size of the image is downsampled by two at each octave. Thus the σ domain is quantised in logarithmic steps, with O octaves and S sub-levels in each octave, as shown in (4) where the base scale is given by σ_0. A typical example of a GSS decomposition is given in Figure 2. Note that a grayscale version of the image is used as the basis for the scale-space.

$$\sigma(o, s) \quad = \quad \sigma_0 \, 2^{(o \, + \, s/S)}, \quad o \in o_{\min} + [0, ..., O-1], \quad s \in [0, ..., S-1] \tag{4}$$

As suggested by its name, the DoG detector then proceeds to derive a further difference of Gaussians scale-space (DoGSS) by taking the difference between successive sub-level images in each octave. Thus if the GSS is given by (5), the DoGSS will be given by (6). Figure 3 shows the DoGSS corresponding to the GSS of Figure 2.

$$G\big(x, y, o, s\big) \quad = \quad S\big(x, y, \sigma(o, s)\big) \tag{5}$$

$$D\big(x, y, o, s\big) \quad = \quad S\big(x, y, \sigma(o, s+1)\big) - S\big(x, y, \sigma(o, s)\big) \tag{6}$$

The Gaussian scale-space representation is in fact the general family of solutions to the diffusion equation

$$\frac{\partial S}{\partial \sigma} = \frac{1}{2} \nabla^2 S$$

thus it follows that the limiting case of the DoGSS will correspond to the Laplacian operator

$$\nabla^2 S\big(x, y, \sigma\big) \approx \frac{S\big(x, y, k\sigma\big) - S\big(x, y, \sigma\big)}{k\sigma - \sigma}.$$

Therefore, optimal points over the DoGSS will approximate the optimal points found by the normalised Laplacian detector of (3).

The SIFT detector obtains an initial set of possible features by finding local optima within the DoGSS for the image. These are obtained coarsely by finding pixels which are greater than (or less than) all neighbouring pixels within the current sub-level and the sub-levels before and after over the DoGSS. Low contrast features are rejected at this stage by requiring the magnitude of potential pixels in the DoGSS to exceed a threshold. These potential features are then presented to the next stage for accurate (sub-pixel) localisation.

2 Feature Point Localisation

The feature point detector provides a list of putative feature points in the DoGSS of the image. These feature points are coarsely localised, at best to the nearest pixel, dependent upon where the

Input Image

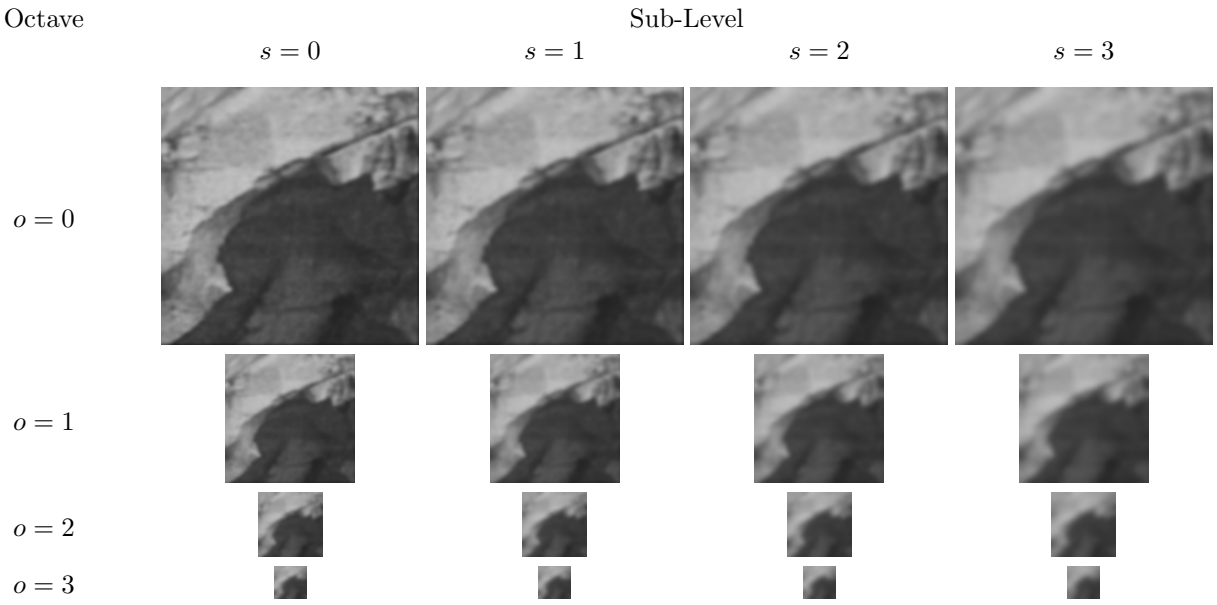


Figure 2: An example of a Gaussian scale-space generated from the input image. *The smoothing is increased from left to right by increasing the variance of the Gaussian kernel which is convolved with the grayscale image. Each level, from top to bottom, is produced by downsampling by two from the previous level before further smoothing.*

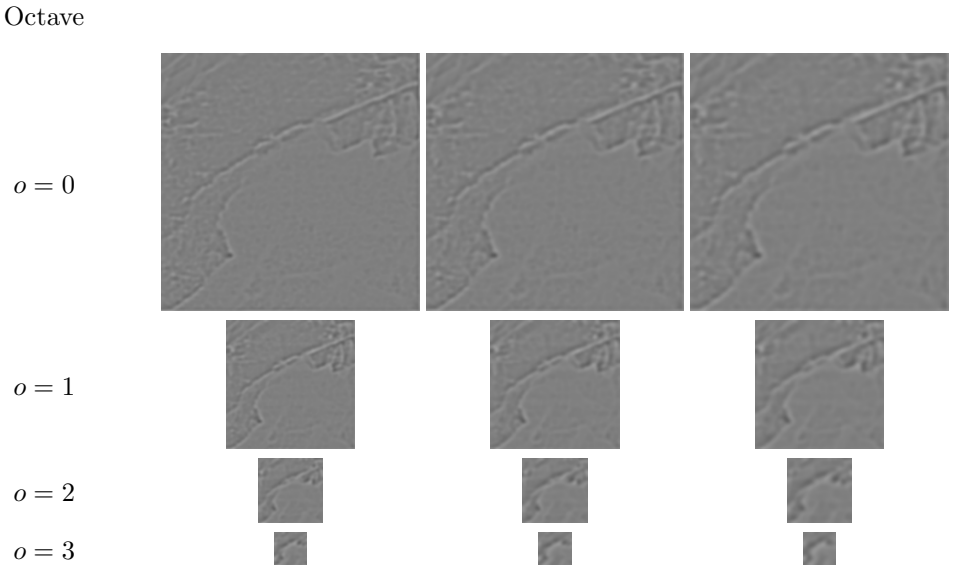


Figure 3: The 'Difference of Gaussians' (DoG) scale-space corresponding to the Gaussian scale-space of Figure 2. *At each level in the Gaussian scale-space, the difference between successive images provides the DoG scale-space.*

features were found in the scale-space. They are also poorly localised in scale since σ is quantised into relatively few steps in the scale-space. The second stage in the SIFT algorithm refines the location of these feature points to sub-pixel accuracy whilst simultaneously removing any poor features.

The sub-pixel localisation proceeds by fitting a Taylor expansion to fit a 3D quadratic surface (in x,y and σ) to the local area to interpolate the maxima or minima. Neglecting terms above the quadratic term, the expansion of the DoGSS of (6) is given in (7) where the derivatives are evaluated at the proposed point $\mathbf{z}_0 = [x_0, y_0, \sigma_0]^T$ and $\mathbf{z} = [\delta x, \delta y, \delta\sigma]^T$ is the offset from this point.

$$D(\mathbf{z}_0 + \mathbf{z}) \approx D(\mathbf{z}_0) + \left(\frac{\partial D}{\partial \mathbf{z}}\bigg|_{\mathbf{z}_0} \right)^T \mathbf{z} + \frac{1}{2}\mathbf{z}^T \left(\frac{\partial^2 D}{\partial \mathbf{z}^2}\bigg|_{\mathbf{z}_0} \right) \mathbf{z} \tag{7}$$

The location of the extremum $\hat{\mathbf{z}}$ is then determined by setting the derivative with respect to $\mathbf{z}$ equal to zero as in (8).

$$\hat{\mathbf{z}} = -\left(\frac{\partial^2 D}{\partial \mathbf{z}^2}\bigg|_{\mathbf{z}_0} \right) \left(\frac{\partial D}{\partial \mathbf{z}}\bigg|_{\mathbf{z}_0} \right) \tag{8}$$

The parameters in (8) may be estimated using standard difference approximations from neighbouring sample points in the DoGSS resulting in a 3×3 linear system which may be solved efficiently. The process may need to be performed iteratively since if any of the computed offset values move by more than half a pixel it becomes necessary to re-evaulate (8) since the appropriate neighbourhood for the approximation will have changed. Points which to not converge quickly are discarded as unstable.

The value at the localised extremum may be interpolated, as in (9), and any points with a value below a certain threshold rejected as low contrast points.

$$D(\hat{x}, \hat{y}, \hat{\sigma}) = D(\mathbf{z}_0 + \hat{\mathbf{z}}) \approx D(\mathbf{z}_0) + \frac{1}{2}\left(\frac{\partial D}{\partial \mathbf{z}}\bigg|_{\mathbf{z}_0} \right)^T \hat{\mathbf{z}} \tag{9}$$

A final test is performed to remove any features located on edges in the image since these will suffer an ambiguity if used for matching purposes. A peak located on a ridge in the DoGSS (which corresponds to an edge in the image) will have a large principle curvature across the ridge and a low one along it whereas a well defined peak will have a large principle curvature in both directions. The Hessian in x and y, given in (10) is evaluated for the feature point, again using a local difference approximation, and the ratio of the eigenvalues λ_1 and λ_2, which correspond to the principle curvatures, compared to a threshold ratio r as in (11) and high ratio points rejected.

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{10}$$

$$\frac{\mathrm{Tr}^2(\mathbf{H})}{\mathrm{Det}(\mathbf{H})} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \lambda_2} < \frac{(r+1)^2}{r} \tag{11}$$

3 Orientation Assignment

The rotation invariance in the SIFT algorithm is provided by determining an orientation for each of the detected features. This is accomplished by investigating the image gradients in the region of the image surrounding the feature. Figure 4 shows the orientation assignment process for the feature

point shown in Figure 4(a). For every localised feature point the nearest image in the GSS is selected, retaining the scale invariance. From this image, pixel difference approximations are used to derive the corresponding gradient image split into magnitude of gradient and angle. Figures 4(c) and 4(d) show the two gradient images generated from the appropriate GSS image in Figure 4(b).

To determine the orientation, a histogram of gradient angles is generated. The gradient angle image determines which orientation bin in the histogram should be used for each pixel. The value added to the bin is then given by the gradient magnitude weighted by a Gaussian window function centred on the feature point, thus limiting to local gradient information. Figure 4(e) gives the Gaussian window function whose extent is determined by the detected scale σ of the feature point. The sub-pixel location and scale is used to improve the accuracy and the resultig histogram is shown in Figure 4(f). The histogram is then smoothed with a moving average filter to produce Figure 4(g). A quadratic fit to the peaks of the smoothed histogram is used to determine the orientations of the feature point. If the histogram has more than one distinct peak then multiple copies of the feature point are generated, each with one of the possible orientations. The yellow line in Figure 4(a) shows the final orientation determined for the feature point under the convention of dark to light regions.

4 Feature Descriptor Generation

The final stage of the SIFT algorithm is to generate the descriptor which consists of a normalised 128 dimensional vector. At this stage of the algorithm we are provided with a list of feature points which are described in terms of location, scale and orientation. This allows us to construct a local co-ordinate system around the feature point which should be similar across different views of the same feature.

The descriptor itself is a histogram formed from the gradient of the grayscale image. A 4×4 spatial grid of gradient angle histograms is used. The dimensions of the grid are dependent on the feature point scale and the grid is centred on the feature point and rotated to the orientation determined for the keypoint. Each of the spatial bins contains an angle histogram divided into 8. The image gradient magnitude and angle are again generated from the scale-space, as in Figures 4(c) and 4(d). The gradient angle at each pixel is then added to the corresponding angle bin in the appropriate spatial bin of the grid. The weight of each pixel is given by the magnitude of the gradient as well as a scale dependent Gaussian centred on the feature point as in Figure 4(e). During the histogram formation tri-linear interpolation is used to add each value, that is interpolation in x, y and θ. This consists of interpolation of the weight of the pixel across the neighbouring spatial bins based on distance to the bin centres as well as interpolation across the neighbouring angle bins. The effect of the interpolation is demonstrated in Figure 5.

The resulting descriptor is then normalised, truncated and normalised again to provide the final vector. Figure 6 shows the descriptor generated for the feature point of Figure 4. The descriptors may be collected from all the feature points in an image and then used for matching between images as necessary.

5 SIFT Feature Matching

Since the SIFT descriptor is a vector in 128 dimensions, Lowe proposes a very simple matching scheme based on the nearest neighbours. Essentially a feature descriptor from the query image may

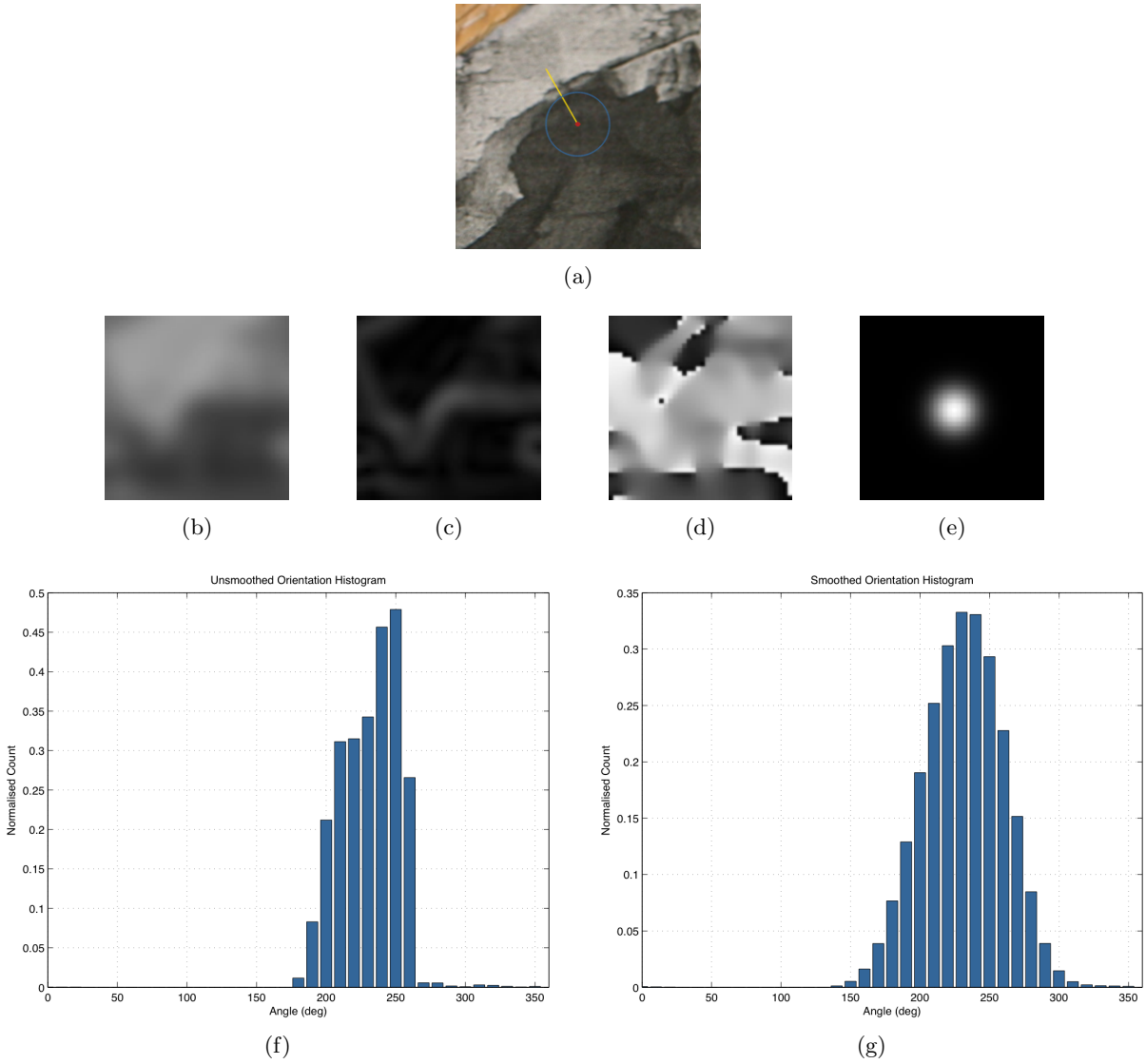
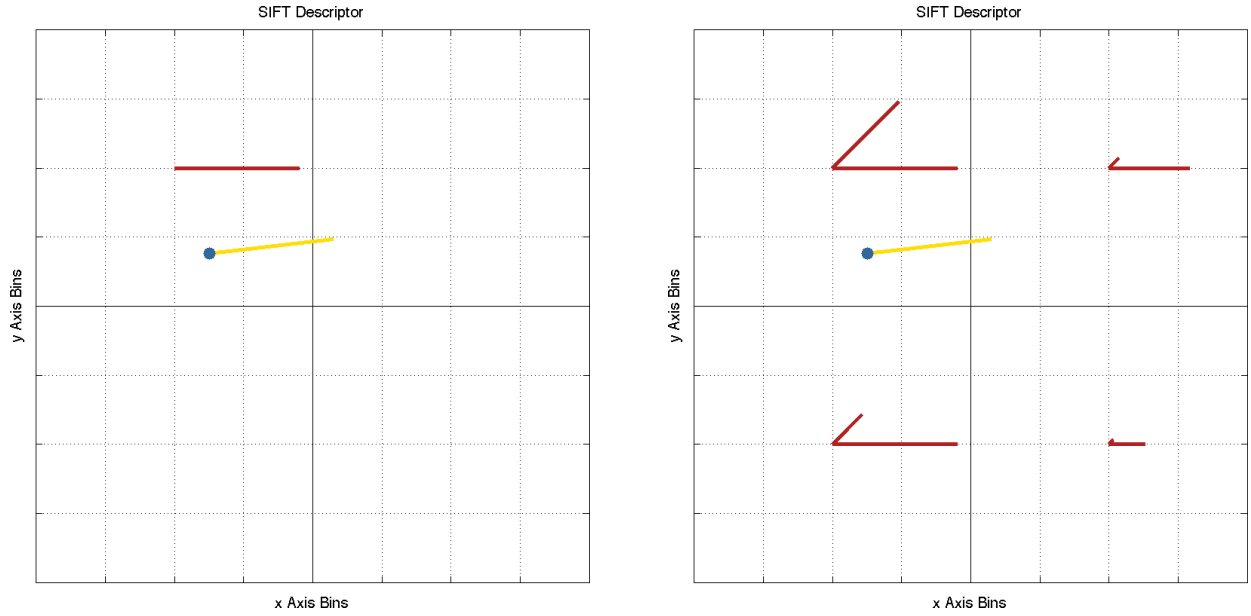


Figure 4: Determining the orientation of a keypoint. *The final orientation (a) is obtained from the appropriate level of the Gaussian scale-space (b) by taking the image gradient, split into magnitude (c) and angle (d), and creating a histogram of the gradient angle using the magnitude weighted by a scale dependent Gaussian (e). The generated histogram (f) is then smoothed (g) and the peaks localised to provide the appropriate angles.*

be compared to all the descriptors of features in the search image and matched to the feature with the closest descriptor vector. The problem with this approach is that there may be features which are not found in both images, therefore the nearest neighbour scheme enforces that a match is always returned, even if the descriptors themselves are not known. Lowe's solution is to compare the descriptors of the two nearest neighbours found in the search image. If the second nearest descriptor differs significantly from the first nearest neighbour then we assume that the descriptor is isolated in the vector space and may therefore be considered a good match, otherwise the match is rejected. This proceeds along the following lines:

$\hat{\mathbf{d}}_q$	is the query descriptor
$\hat{\mathbf{d}}_1$	is the first nearest descriptor in the search image
$\hat{\mathbf{d}}_2$	is the second nearest descriptor in the search image

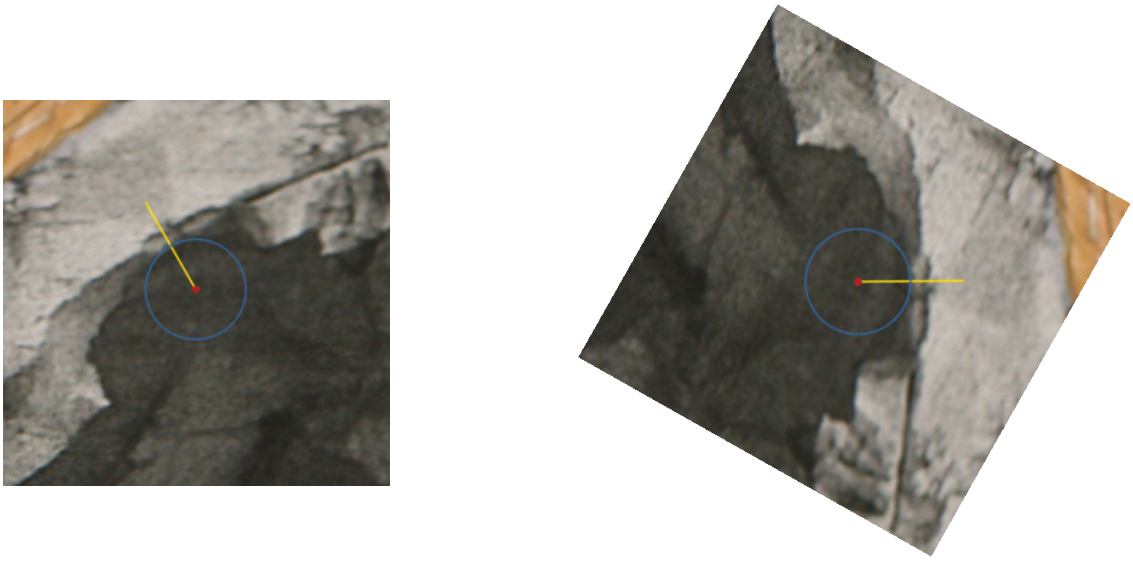


(a) Descriptor without Interpolation (b) Descriptor with Interpolation

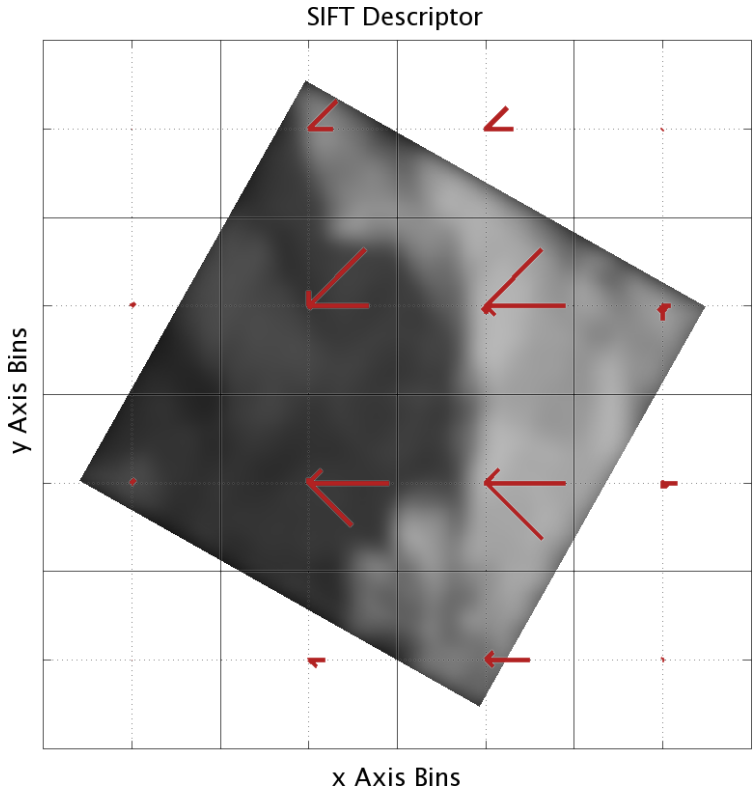
Figure 5: The effect of interpolation when generating the SIFT descriptor. *A single pixel's gradient angle is added to the descriptor. Without interpolation the full weight will only be added to a single bin (a) whereas when tri-linear interpolation is used (b) the weight is spread across neighbouring spatial bins and across the corresponding neighbouring angle bins. This is advantageous since it is more robust for matching different images when there may be differences in the localisation and orientation of the same feature in the different views.*

$$\text{reject unless} \quad \frac{\cos^{-1}(\mathbf{d}_q \cdot \mathbf{d}_1)}{\cos^{-1}(\mathbf{d}_q \cdot \mathbf{d}_2)} < r \quad \text{where } r \text{ is the threshold ratio.}$$

Figure 7 shows the results of this matching technique applied to the query and search images of Figure 1 with a threshold of $r = 0.6$. The matching results may return erroneous correspondences (outliers) in addition to correct matches (inliers) and therefore further processing stages are required to robustly determine the inlier matches. This allows the outliers to be ignored during the subsequent estimation of correlations between the images.



Input Image Rotated Image



SIFT Descriptor for Keypoint

Figure 6: The SIFT descriptor generated for the feature point in Figure 4. *The SIFT descriptor is generated by forming a weighted histogram of the image gradient angle around the feature point. A* 4×4 *spatial grid is centred on the feature point at the correct orientation and scale. Each spatial bin contains 8 angle histogram bins (similar to the histogram used to determine orientation in Figure 4). The image gradient angles are added to the appropriate angle bin in the appropriate spatial bin using tri-linear interpolation (interpolation across the spatial bins and the angle bins) in order to increase robustness to slight differences in orientation and registration across different matching images. As for the orientation stage, the histogram values are weighted by a scale dependent Gaussian centred on the feature point. The final output is a normalised vector in 128 dimensions.*

Figure 7: The initial SIFT feature matches. *The SIFT features for the two images are matched against one another using a rejection threshold ratio of 60%. The putative correspondences show that there are some erroneous matches (outliers) including matches to the corners of different posters in the search image.*